

Cairo University

Faculty of Engineering

Department of Computer Engineering

**LA**خ**SLY**



A Graduation Project Report Submitted

to

Faculty of Engineering, Cairo University

in Partial Fulfillment of the requirements of the degree

of

Bachelor of Science in Computer Engineering.

**Presented by**

Ahmed Salama Hamed Ahmed Maher

Moamen hassan attia Mohamed Talaat Mohamed

**Supervised by**

Dr. MagdaFayek

2020

All rights reserved. This report may not be reproduced in whole or in part, by photocopying or other means, without the permission of the authors/department

**General Guidelines**

* The color of the cover page should be green
* Throughout your text, use the font type, font size, and spacing, as in this template. In general, Arial fonts should be used. Chapter headings should be of size 24. Sections should be of size 18, and regular text should be of size 12. Your text should be justified on both left and right sides.
* The reference list should be written using a font size of 10. Ensure that the references are written correctly and all fields are included. References should be ordered according to their appearance in the text “[1], [2], [3] … etc”
* The table of content is a tentative one. You could add more sections as required. However, the mentioned sections should be included in your report
* For the appendices, add any appendix you see necessary. Remove any appendix that is not applicable to your project. However, the feasibility study and user guide should be included
* Ensure that the report is clear and self-contained, such that any future interested reader could completely understand your project “to the extent of building another one similar to yours”
* Any figure caption should be inserted below the figure. Figures within any chapter should be numbered starting from 1. For example, the first figure of chapter 2 should be “Figure 2.1”. Similarly, the fourth figure of chapter 3 should be “Figure 3.4”
* Any table caption should be inserted above the table. Tables within any chapter should be numbered starting from 1. For example, the first table of chapter 4 should be “Table 4.1”. Similarly, the seventh table of chapter 5 should be “Table 5.7”
* Copy and paste from any other source is not allowed by any shape. Even for the background knowledge, you have to use your own wording.
* The complete report should be submitted 48 hours before the final project demonstration day. Ensure that you would meet this deadline to avoid any late penalty

# Abstract

Sports events attract audiences from all over the world especially in recent years and football is the most popular out of these sports. Sports in general and specifically football matches consist of some interesting and important events such as goals, dangerous attacks, fouls and red cards. and other events that are less interesting such as dull passes in the middle of the football pitch. So due to the recently increased amount of sports video content, summarizing these interesting events from the sport video into a much shorter video can greatly enhance the sports fan experience and save a lot of time. Therefore, in recent years football video summarization and analysis has attracted many researchers and a lot of applications.

Football video summarization is a process of selecting the most interesting events that happen in a match to produce a shorter summary of that match by using various techniques such as processing audio, image, and text information from the original video.

In this report we present our approach in the field of football summarization using different techniques in different fields as image processing, and machine learning. Our proposed system mainly consists of five phases. First the preprocessing phase where we divide the video into patches of frames to be processed separately avoiding overhead. Second the shot boundary phase where shot boundary detection is used on every patch of frames to separate the video into shots by finding the transitions between each camera to another. Then each of these shots is processed by the next three phases, the third phase is the shot classification phase where each shot is classified to a specific type; wide, medium, close, close-out, and logo. Fourth is the event detection phase where to determine whether the shot contains a goal, it has a high level of audio volume, or it contains a goal-mouth. Finally, in the summarization phase where the output of all of the previous phases is concluded and analyzed to decide which of the shots is interesting enough to keep it in the output video and which one can be discarded to finally produce the summarized video.

# الملخص

تجذب الأحداث الرياضية جماهير من جميع أنحاء العالم خاصة في السنوات الأخيرة وتعتبر كرة القدم هي الأكثر شعبية من بين هذه الألعاب الرياضية. تتكون الرياضة بشكل عام ومباريات كرة القدم على وجه التحديد من بعض الأحداث المثيرة للاهتمام والمهمة مثل الأهداف والهجمات الخطيرة والخطأ والبطاقات الحمراء. وأحداث أخرى أقل إثارة للاهتمام مثل التمريرات الباهتة في منتصف ملعب كرة القدم. لذلك، نظرًا لزيادة كمية محتوى الفيديو الرياضي مؤخرًا، فإن تلخيص هذه الأحداث المثيرة للاهتمام من الفيديو الرياضي إلى فيديو أقصر بكثير يمكن أن يعزز بشكل كبير تجربة المعجبين الرياضيين ويوفر الكثير من الوقت. لذلك، في السنوات الأخيرة، جذب تلخيص وتحليلات فيديو كرة القدم العديد من الباحثين والكثير من التطبيقات.

تلخيص فيديو كرة القدم هو عملية اختيار الأحداث الأكثر إثارة للاهتمام التي تحدث في مباراة لإنتاج ملخص أقصر لتلك المباراة باستخدام تقنيات مختلفة مثل معالجة الصوت والصورة ومعلومات النص من الفيديو الأصلي.

تقدم هذه الرسالة نظام تلخيص اوتوماتيكي لفيديو كرة القدم باستخدام مجموعة متنوعة من التقنيات في مجالات معالجة الصور، وتعلم الآلة. يتكون النظام المقترح من خمس مراحل. أولا، مرحلة ما قبل المعالجة، يقرأ النظام الفيديو في شكل صور منفردة. ثانيا، يتم تطبيق مرحلة تحديد مدة اللقطة على كل مجموعة من الصور لتقسيم الفيديو إلى لقطات منفصلة، أي العثور على الانتقال من كاميرا إلى أخرى. ثم يتم معالجة كل لقطة بشكل منفصل في المراحل الثلاث التالية، مرحلة تصنيف اللقطة، معنية بتصنيف كل لقطة على سبيل المثال اللقطة الواسعة عندما تكون رقعة ملعب كرة القدم مرئية بالكامل، اللقطة القريبة عندما تكون وجوه اللاعبين مرئية وما إلى ذلك. ثم مرحلة اكتشاف احداث المباراة والتي يتم تحليل كل لقطة فيها لتحديد ما إذا كان هناك هدف يحدث أم لا أو إذا كان مستوى الصوت في هذه اللقطة مرتفعًا مقارنة بالفيديو بأكمله أو إذا ظهر المرمي في اللقطة. وأخيرًا، مرحلة التلخيص التي يتم فيها تحليل مخرجات جميع المراحل السابقة لتحديد أي لقطة مهمة ويجب تضمينها في الفيديو النهائي وأي اللقطات يجب تجاهله لصنع الملخص.

# ACKNOWLEDGMENT

We want to express our gratitude to Dr. Magda Fayek for encouraging us and motivating us to work on this project and gain so much valuable experience. We also want to thank our friends and family for supporting us and before all thanks to Allah for his blessings.

# Table of Contents

[Abstract](#_gjdgxs) 3

[الملخص](#_30j0zll) 4

[ACKNOWLEDGMENT](#_1fob9te) 5

[Table of Contents](#_3znysh7) 6

[List of Figures](#_3dy6vkm) 8

[List of Tables](#_4d34og8) 9

[List of Abbreviation](#_2s8eyo1) 10

[List of Symbols](#_17dp8vu) 11

[Contacts](#_3rdcrjn) 12

**[Chapter 1: Introduction](#_26in1rg)**

[1.1. Motivation and Justification](#_lnxbz9) 14

[1.2. Project Objectives and Problem Definition](#_35nkun2) 14

[1.3. Project Outcomes](#_1ksv4uv) 14

[1.4. Document Organization](#_44sinio) 15

**[Chapter 2: Market Feasibility Study](#_2jxsxqh)**

[2.1. Targeted Customers](#_z337ya) 16

[2.2. Market Survey](#_3j2qqm3) 16

[2.2.1. FilGoal and YallaKora](#_1y810tw) 16

[2.2.2. Youtube Channels](#_4i7ojhp) 16

[2.3. Business Case and Financial Analysis](#_2xcytpi) 17

[2.3.1. Business Case:](#_1ci93xb) 17

[2.3.2. Financial Analysis:](#_3whwml4) 17

**[Chapter 3: Literature Survey](#_qsh70q)**

[3.1. Background on Video structure](#_3as4poj) 20

[3.2. Background on color Spaces](#_49x2ik5) 20

[3.3. Background on Convolutional Neural Networks (CNN)](#_3o7alnk) 21

[3.4. Comparative Study of Previous Work](#_ihv636) 22

[3.5. Implemented Approach](#_32hioqz) 23

**[Chapter 4: System Design and Architecture](#_1hmsyys)**

[4.1. Overview and Assumptions](#_41mghml) 24

[4.2. System Architecture](#_2grqrue) 24

[4.2.1. Block Diagram](#_vx1227) 26

[4.3. Pre-processing phase](#_1v1yuxt) 27

[4.3.1. Functional Description](#_4f1mdlm) 27

[4.3.2. Modular Decomposition](#_2u6wntf) 27

[4.3.3. Design Constraints](#_3tbugp1) 27

[4.4. Shot Boundary](#_28h4qwu) 28

[4.4.1. Functional Description](#_nmf14n) 28

[4.4.2. Modular Decomposition](#_46r0co2) 28

[4.4.3. Design Constraints](#_4k668n3) 34

[4.5. Excitement Event detection](#_2zbgiuw) 34

[4.5.1. Audio Processing](#_sqyw64) 34

[4.5.2. Goal-Mouth Detection](#_4bvk7pj) 36

[4.5.3. Goal detection](#_25b2l0r) 39

[4.5.3.1. Goal detection using OCR](#_34g0dwd) 39

[4.5.3.2. Goal Detection with structural similarity image index (SSIM)](#_2iq8gzs) 41

[4.6. shot classification](#_3hv69ve) 43

[4.6.1. Shot classification using Grass color ratio](#_2afmg28) 44

[4.6.2. Shot classification using Face detection](#_48pi1tg) 46

[4.6.3. Shot Classification using deep learning](#_2nusc19) 47

[4.6.4. Shot Classification using image processing and machine learning](#_3mzq4wv) 48

[4.6.4. Comparison between shot classification approaches](#_319y80a) 51

[4.7. Summarization and Classification phase](#_40ew0vw) 52

[4.7.1. Functional Description](#_2fk6b3p) 52

[4.7.2. Modular Decomposition](#_upglbi) 52

[4.7.2.1. Extracting Important Events](#_3ep43zb) 52

[4.7.2.2. Shots Sequence Classification](#_184mhaj) 56

[4.8. Audio version of LAخSLY](#_36ei31r) 58

[4.8.1. Reason to develop this version](#_1ljsd9k) 58

[4.8.2. Advantages and disadvantage](#_45jfvxd) 58

[4.8.3. Methodology](#_zu0gcz) 59

**[Chapter 5: System Testing, Verification and Results](#_4iylrwe)**

[5.1. Testing Setup](#_2y3w247) 62

[5.2. Testing Plan and Strategy](#_1d96cc0) 62

[5.2.1. Module Testing and Results](#_3x8tuzt) 63

[5.2.1.1 Shot boundary testing and results](#_2ce457m) 63

[5.2.1.2. Goal-Mouth testing and results](#_3bj1y38) 63

[5.2.1.3. Shot Classification using Deep Learning approach testing and results](#_4anzqyu) 64

[5.2.1.4. Goal Detector using SSIM approach](#_14ykbeg) 64

[5.2.1.5. Audio version testing and results](#_243i4a2) 65

[5.2.2. Integration Testing](#_1idq7dh) 67

[5.3. Testing Schedule](#_1vsw3ci) 69

[5.4. Comparative Results to Previous Work](#_4fsjm0b) 69

**[Chapter 6: Conclusions and Future Work](#_2uxtw84)**

[6.1. Faced Challenges](#_1a346fx) 70

[6.2. Gained Experience](#_2981zbj) 71

[6.3. Conclusions](#_odc9jc) 72

[6.4. Future Work](#_38czs75) 72

[References](#_1nia2ey) 74

[Appendix A: Development Platforms and Tools](#_47hxl2r) 74

[A.1. Hardware Platforms](#_2mn7vak) 74

[A.2. Software Tools](#_11si5id) 74

[Appendix B: Use Cases](#_3ls5o66) 75

[Appendix C: User Guide](#_4kx3h1s) 76

[Appendix D: Feasibility Study](#_302dr9l) 76

# List of Figures

[Figure 2-1 Financial Analysis](#_2bn6wsx) 19

[Figure 3-1 Video Structure](#_1pxezwc) 20

[Figure 3-2 RGB color space](#_2p2csry) 21

[Figure 3-3 HSI color space](#_147n2zr) 21

[Figure 3-4 CNN architecture](#_23ckvvd) 22

[Figure 4-1 LAخSLY Block Diagram](#_3fwokq0) 26

[Figure 4-2 Frames of video](#_19c6y18) 27

[Figure 4-3 example of an instant cut](#_37m2jsg) 28

[Figure 4-4 example of logo transition](#_1mrcu09) 28

[Figure 4-5 example of consecutive frames from the same shot](#_2lwamvv) 30

[Figure 4-6 example of consecutive frames with a transition(cut) between them](#_111kx3o) 31

[Figure 4-7 example of two frames are consecutive but in the midst of a logo transition](#_3l18frh) 31

[Figure 4-8 Flow Chart of Shot Boundary](#_206ipza) 33

[Figure 4-9 Flow chart of the audio processing](#_1rvwp1q) 35

[Figure 4-10 Examples of important events containing goal post](#_2r0uhxc) 36

[Figure 4-11 Flow Chart of goal-mouth](#_3q5sasy) 38

[Figure 4-12 Examples of Scoreboards](#_kgcv8k) 39

[Figure 4-13 Example Before and after a goal](#_1jlao46) 39

[Figure 4-14 Flow Chart of Goal detection using OCR](#_43ky6rz) 40

[Figure 4-15 Flow Chart of Goal detection using SSIM](#_xvir7l) 42

[Figure 4-16 Types of shots](#_1x0gk37) 43

[Figure 4-17 Different tones of green color](#_pkwqa1) 44

[Figure 4-18 example of grass color algorithm results](#_39kk8xu) 45

[Figure 4-19 LeNet architecture](#_1302m92) 48

[Figure 4-20 Golden Section spatial composition rule](#_2250f4o) 48

[Figure 4-21 Flow Chart of shot classification](#_haapch) 50

[Figure 4-22 example of replay detection](#_1tuee74) 52

[Figure 4-23 Flow chart of Extracting Important Events using replay detection](#_4du1wux) 54

[Figure 4-24 Flow chart of Extracting Important Events using audio processing](#_2szc72q) 55

[Figure 4-25 Flow of Classifying shots that come from replays](#_3s49zyc) 56

[Figure 4-26 Flow of Classifying shots that are not replayed but contain high volume level](#_279ka65) 57

[Figure 4-27 Flow Chart of audio processing](#_1yyy98l) 61

[Figure 5-1 chart showing the average score for each league in each audio threshold level](#_338fx5o) 67

[Figure A-1 Use Cases Diagram](#_20xfydz) 76

# List of Tables

[Table 4-1 Comparison between shot classification approaches](#_1gf8i83) 51

[Table 4-2 Advantages and disadvantage of Audio version of LAخSLY](#_2koq656) 58

[Table 5-1 Shot boundary testing and results](#_rjefff) 63

[Table 5-2 Goal-Mouth testing and results](#_1qoc8b1) 64

[Table 5-3 Shot Classification using Deep Learning approach testing and results](#_2pta16n) 64

[Table 5-4 Goal Detector using SSIM approach testing and results](#_3oy7u29) 65

[Table 5-5 Audio version testing and results](#_j8sehv) 66

[Table 5-6 Integration testing results of detailed version](#_42ddq1a) 68

[Table 5-7 Overall results of Integration testing of detailed version](#_2hio093) 68

[Table 5-8 Integration testing results of audio version](#_wnyagw) 69

[Table 5-9 Overall results of Integration testing of audio version](#_3gnlt4p) 69

# List of Abbreviation

BSD Berkeley Source Distribution

Capex Capital Expenditure

CNN Convolutional Neural Networks

CPM Cost per thousand

FC Fully Connected

HSI Hue, Saturation, and Intensity color space

MSE Mean square Error

OCR Optical character recognition

OpenCV Open Source Computer Vision Library

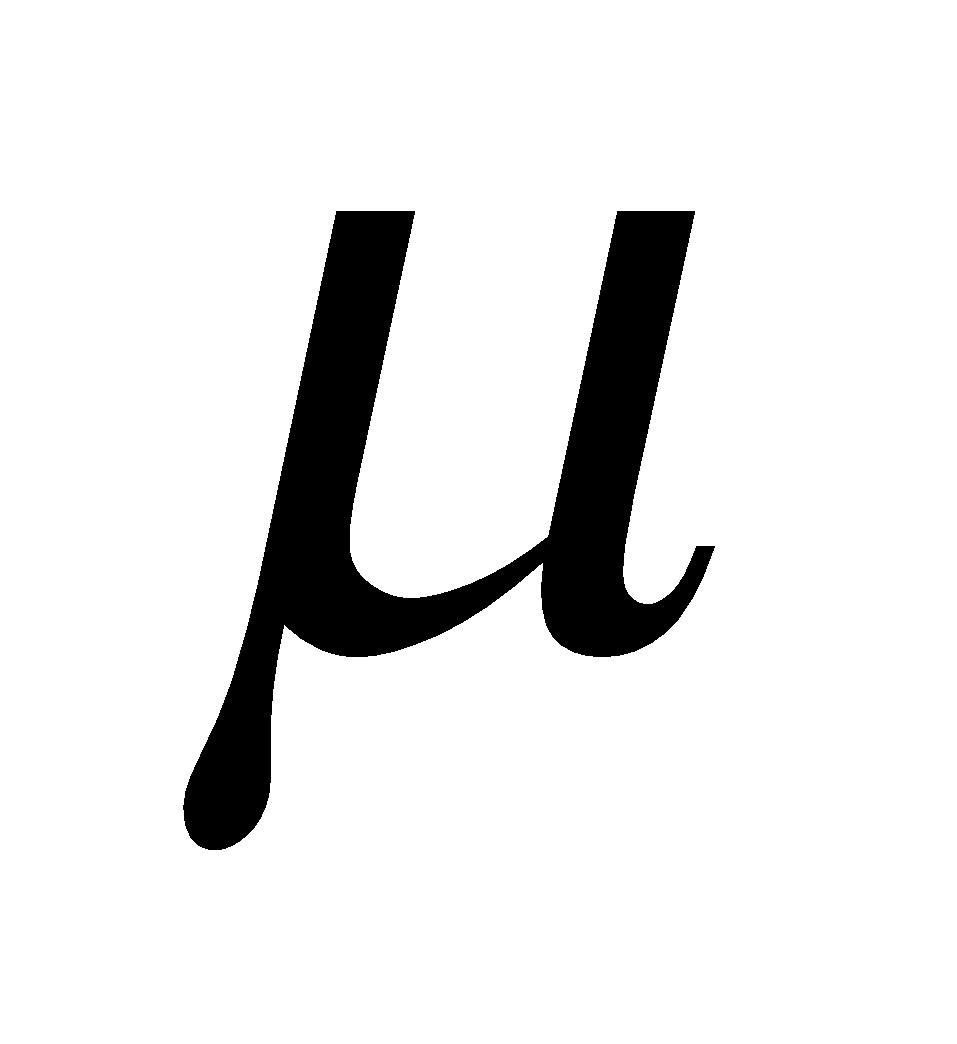
Opex Operational Expenses

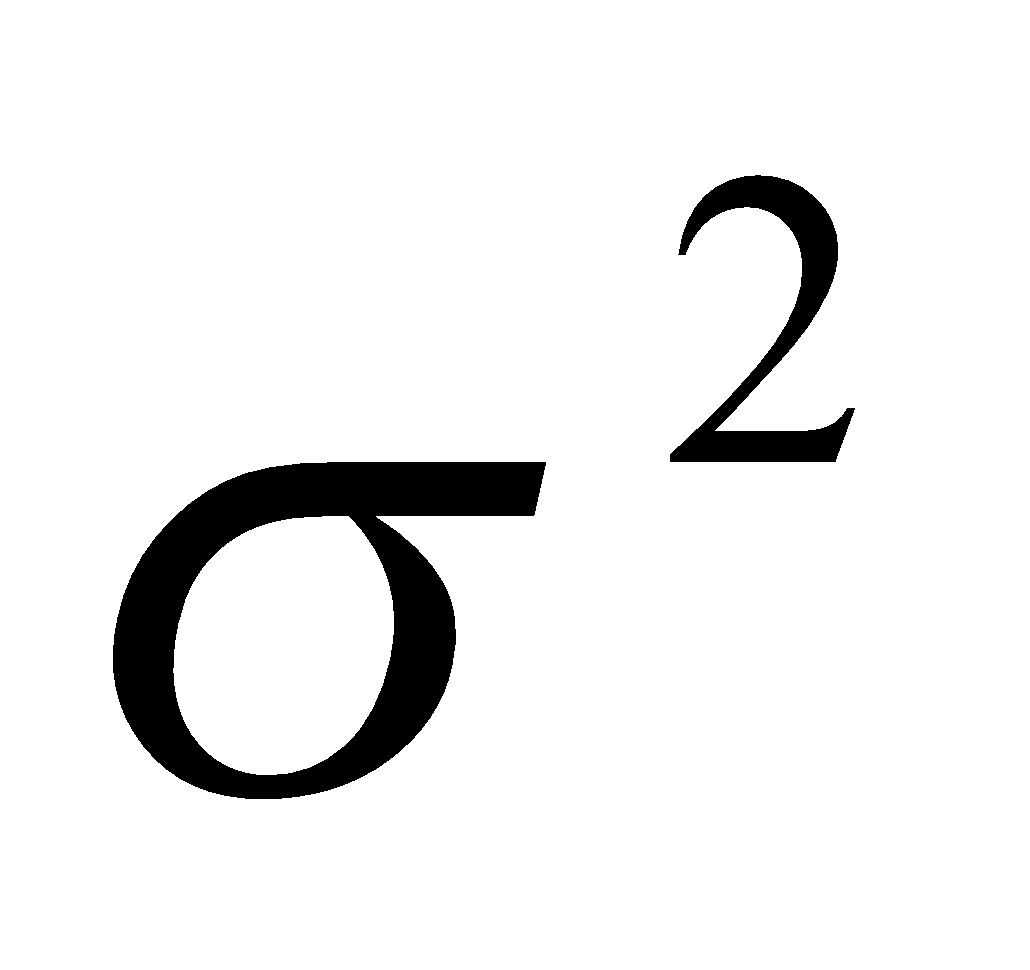
PSNR Peak signal-to-noise ratio

RGB Red, Green, and Blue color space

SSIM Structural similarity image index

# List of Symbols

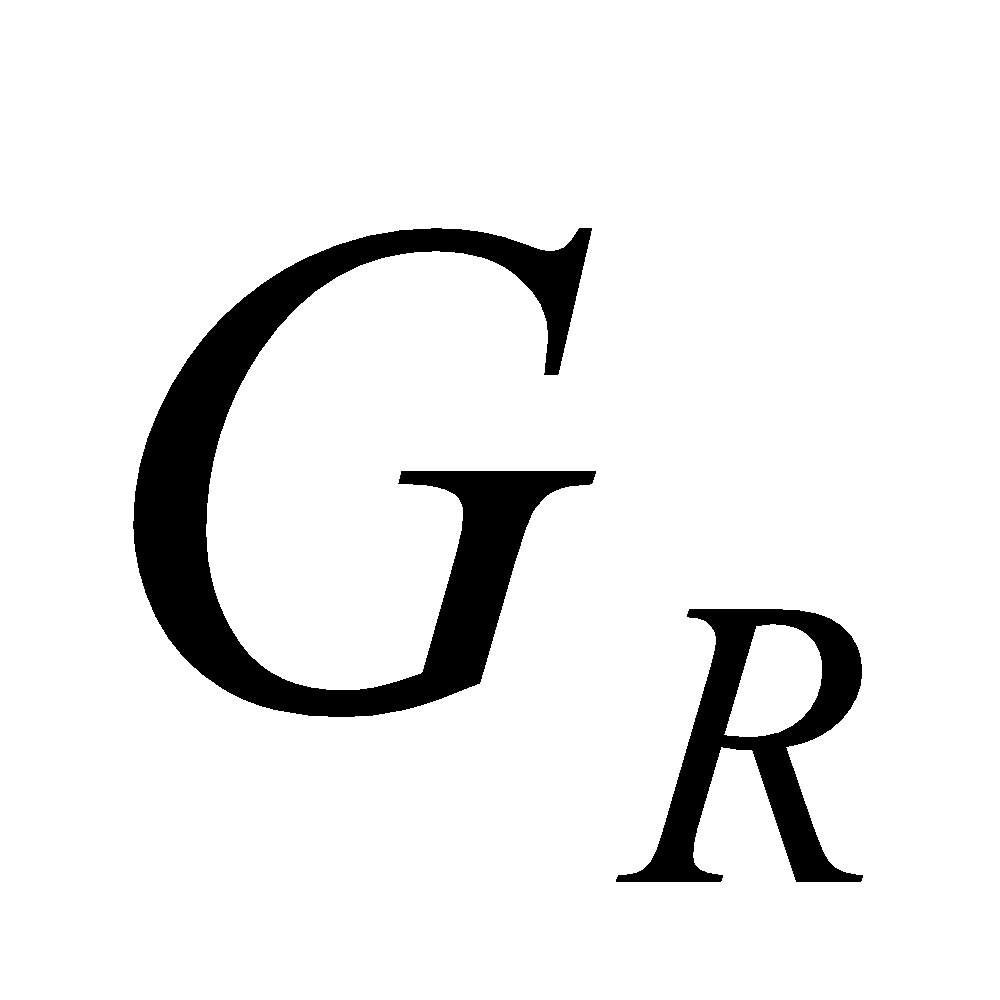
 Average

 Variance

E1 no. Of Goals detected / total goals

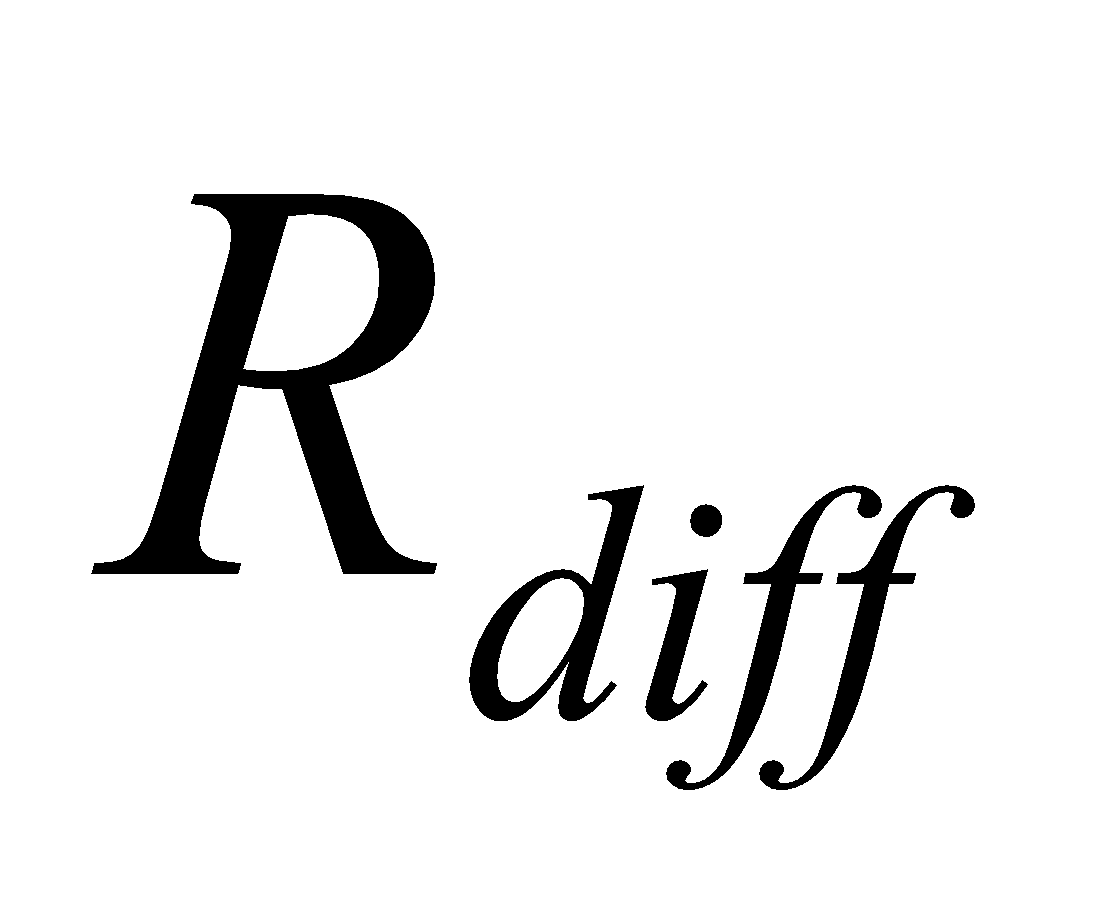
E2 no. of other interesting events detected / total no. Of events

E3 Length of output video / total length of video (=90)

 The grass colored pixel ratio

H Histogram of an image

L The [ratio](https://en.wikipedia.org/wiki/Ratio) between the largest and smallest values that a certain quantity can assume

 The mean value of the absolute grass color pixel differences between regions

# Contacts

**Team Members**

|  |  |  |
| --- | --- | --- |
| **Name** | **Email** | **Phone Number** |
| Ahmed Salama Hamed | [asivo.ahmed@gmail.com](mailto:asivo.ahmed@gmail.com) | +2 01144697687 |
| Ahmed Maher | [demha.ahmed@yahoo.com](mailto:demha.ahmed@yahoo.com) | +2 01065725338 |
| Moamen Hassan Attia | moamenattia@outlook.com | +2 01200370750 |
| Mohamed Talaat Mohamed | [mohamedtalaat0111790@gmail.com](mailto:mohamedtalaat0111790@gmail.com) | +2 01115598525 |

**Supervisor**

|  |  |  |
| --- | --- | --- |
| **Name** | **Email** | **Number** |
| Dr. Magda Fayek | [magdafayek@gmail.com](mailto:magdafayek@gmail.com) | +2 01xxxxxxxxx |

This page is left intentionally empty

# Chapter 1: Introduction

In this chapter we introduce the important value of football analysis and summarization as football is a major sport that attracts a large audience. The fact that the number of contents has been increasing so much recently, watching the full football match can consume a lot of time, so many football fans tend to watch a summarized video of the highlights of the match instead. Discussing the problems that face video summarization especially in football and presenting our proposed system that tries to solve them. Finally, a conclusion and an overview about the proposed system at the end of the chapter.

## Motivation and Justification

There are many reasons which motivated us to make Laخsly but the main two are:

* Most football fans follow multiple teams but they cannot watch all the games that their favorite teams play all year and matches are played at the same time within different time zones.
* Coaches need to view the important events in their matches and their opponents' matches to truly develop and evaluate the team players and their performance.

It would entertain fans and popularize the sport itself if the fans could watch many more games without spending much time So, football analysis is important and worth the effort that goes into research and development.

## Project Objectives and Problem Definition

The objective is to input a football match to be summarized by a computer program, the program should extract the most exciting events in the football game such as (goals, attacks, and other events) then output those events into a summarized video.

In this report, we present a system for football summarization called LAخSLY and discuss its inner workings and try to improve on the research that has been done in the field of summarization of sports.

## Project Outcomes

After the preprocessing phase the frames of the video go through the phases of the system to be analyzed. The shot boundary detection phase segments the input video stream into small video shots by applying a combination of image processing techniques and histogram comparison techniques. The shot classification phase applies different algorithms, namely; Grass color ratio, Face Detection, Deep learning and image processing techniques to classify the shot into one of the following classes: wide – medium - close – close out – logo. In The event detection phase each shot is considered separately and compared with the previous shot to determine if a goal has happened between the two shots by comparing the scoreboard in the upper left corner. Two methods are proposed to detect a goal event, Optical character recognition (OCR) and structural similarity image index (SSIM). Lastly, the summarization phase which takes into account the output of the previous phases to decide if the shot is important and should be included in the final video or should be discarded. For example, if a shot contains a goal then it is important or if a shot has a goal mouth and the audio level is high then it is important or it’s a replay shot and a replay of course is for something important. Then combining the important shots only in a video and that is the output of the system

## Document Organization

In chapter 2, we discuss the competitors to laخsly and study the market to understand how a football summarization system would compete in today’s market and also perform financial analysis to estimate costs and predict profits.

In chapter 3, we review the background on how a generic video is structured and discuss color spaces and how color is represented in digital images to be able to properly understand the processing that happens in the following chapters and also discuss the efforts done in the field of football summarization.

In chapter 4, we present the architecture of the proposed system and discuss each phase of the five phases of the system in detail.

In chapter 5, we present a faster version of laخsly that depends on the audio feature only and the shot boundary process and discuss why developing this version is worth the effort, it’s advantages and disadvantages.

In chapter 6, we discuss how we approached testing the completed system as well as each module individually and present verification for each module and our final results.

In chapter 7, we draw our conclusions and present ways that could expand the project even more in the future.

# 

# 

# Chapter 2: Market Feasibility Study

There are many platforms where a user can watch a summary of a football match such as “YouTube”, “Filgoal” and “YallaKora”, but in these platforms the summaries are produced manually, and even some online platforms provide only datasets for action spotting in football but do not provide a summarized video of these actions.

A system for summarizing may be available to corporations or clubs but there isn’t a product that is available for the mass public to use and that was an additional motivation for embarking on this project.

## Targeted Customers

As mentioned before such a system’s output is intended for the dedicated sports fans especially football fans who are not able to keep up with all the matches, they want to watch so they resort to watching summaries of the matches they miss on such platforms such as YouTube.

## Market Survey

The football summarization market is not scarce by any means, there is a plethora of options available for the user if he/she wants to watch a summary of a certain match but these options mostly depend on manual labor to produce summaries. Below we discuss examples of said available football summary sources and their market position with respect to other competitors.

### FilGoal and YallaKora

These two platforms are very similar in the service they provide to the audience, they both act as online journals that provides audience by football news all over the world every day about all leagues and championships, Although they provide news and information about other sports but their main focus is on football, for every match, they provide the events happening in the match minute by minute, they provide a video for each scored goal, finally, after the match ends, they provide a video of all of the scored goals.

The main advantages of such platforms are:

* Provide football audience by the updated news about football all over the world
* Make their audience feel as if they were watching the match live by sending updates by events happening each minute in the match.
* Provide a video containing goals scored after each match.

The main disadvantages are:

1. Do not provide a video summary of the highlights of the match (only goals).
2. All of the services provided to the audience are done manually, which consumes a lot of money and time.

### YouTube Channels

Many YouTube channels that are specialized in football provide a neat and clean summary after each match containing all the highlights and goals in the match.

These channels are the main competitor to the proposed project as they provide exactly the same service and for free.

The main advantages are:

* Provides a neat and clean match summary
* Free

The main disadvantages are:

* Done manually, which consumes time and money.
* Most of the channels do not have the rights to share the videos online, which causes them to get copyright strikes.

## Business Case and Financial Analysis

### Business Case:

* The proposed service will be provided to the audience as a website serving the summarized video of matches, and the user can search for any match he/she wants and watch its summary.
* The service will be available for free to be a strong competitor to the service provided by the “YouTube” channels, and will be dedicated to football video summarization to be a strong competitor to “FilGoal” and “YallaKora”.
* The main source of revenue will be based on online advertisement that will be presented on the website, there are many ways to make money with a website such as, “Affiliate Marketing”, “Pay Per Click” advertising, “Sell ad space”, “Sell sponsored posts”,etc.
* Over the next 5 years the revenues will mainly depend on the number of visitors of the website per month, the more the website visitors the more the advertisements in the website, and this means more money.

### Financial Analysis:

* **Capex** (Capital Expenditure):
* Starting by 4 developers in the company, 4 laptops = 15,000 EGP/item, 4 desks = 2,500 EGP/item and 4 chairs = 1,500 EGP/item are needed.
* One air conditioner in the development room and another in the meeting room

= 10,000 EGP/item

* **Opex** (Operational Expenses):
  + Renting an apartment with two rooms (development room and meeting room), a kitchen and a bathroom = 7,000 EGP/month
* Salaries = 8,000 EGP/month for each developer.
* Marketing (YouTube advertisement) = 5$/day = 2400 EGP/month
* Renting servers (storage: 100 GB, RAM: 4 GB, speed = 100 Mbps)

= 550 EGP/month

* Power of attorney = 930 EGP/month
* Name clearance certificate = 750 EGP/month
* **Revenue**
  + Will be based on online advertisement using “Sell ad space” method and will be calculated by how many visitors the website gets.
  + Typically, it’s quoted as a dollar amount per one thousand impressions (or CPM). On average it is $5 = 80 EGP CPM. If the website gets 100,000 visits a month, that ad price translates into $500 bucks = 8000 EGP.
  + The good thing about this approach is that if the site gets a ton of traffic from different sources, a simple banner ad pricing can go up to as high as $5000 = 80,000 EGP per month! The obvious downside is that if the site doesn’t get a lot of traffic, you can’t expect to earn much either.
  + Assuming that the website will get 500,000 visits/month = 40,000 EGP/month in the first year and the number will double every year.

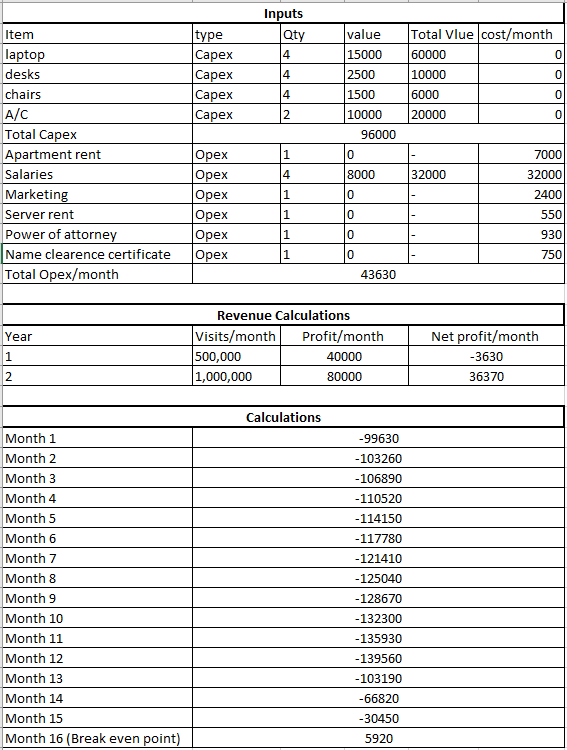
****

Figure 2-1 Financial Analysis

# 

# 

# Chapter 3: Literature Survey

In this chapter we review the background on how a generic video is structured and discuss color spaces and how color is represented in digital images to be able to properly understand the processing that happens in the following chapters and also discuss the efforts done in the field of football summarization.

## 3.1. Background on Video structure

To have a good understanding of each phase in our system, first we must discuss how the video is structured.

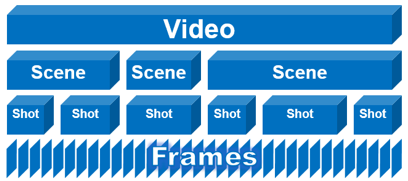


Figure 3-1 Video Structure

As shown in the above figure, the video consists of a collection of scenes which are combined of shots which each shot is combined of frames. Scenes can be broken to shots due to different editing techniques, camera angles, views, and transitions between scenes. Processing the video frame by frame will be very overhead for both the processing power of the device and the time, also processing scenes can affect certain fine details or events that can be interesting. Therefore, shots are considered the main unit for video analysis.

## 

## 

## 

## 

## 3.2. Background on color Spaces

Various color spaces exist because they present color information in ways that make certain calculations more convenient or because they provide a way to identify colors that are more intuitive. For example, the RGB color space defines a color as the percentages of red, green, and blue hues mixed together. Other color models describe colors by their hue (shade of color), saturation (amount of Gray or pure color), and luminance (intensity, or overall brightness).

The RGB color space represents images as an m-by-n-by-3 numeric array whose elements specify the intensity values of the red, green, and blue color channels. The range of numeric values depends on the data type of the image but normally it’s in the range 0-255.

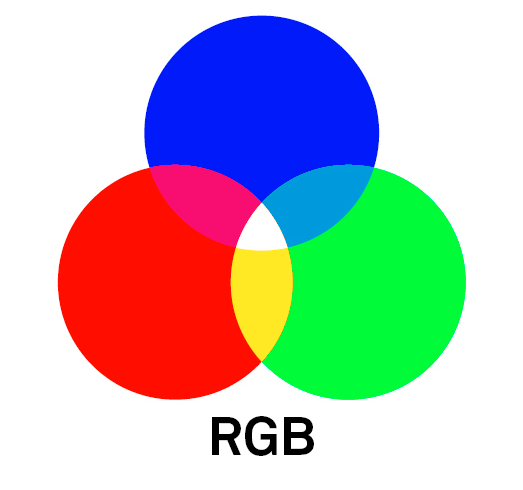


Figure 3-2 RGB color space

The HSI color space is a very important and attractive color model for image processing applications because it represents colors similarly to how the human eye senses colors.

The HSI color model represents every color with three components: hue(H), saturation(S), intensity(I). The below figure illustrates how the HIS color space represents colors.



Figure 3-3 HSI color space

## 3.3. Background on Convolutional Neural Networks (CNN)

In deep learning, a convolutional neural network (CNN, or Convent) is a class of deep neural networks. They are mostly applied to applications in image and video recognition, image classification and natural language processing.

CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

A CNN is comprised of several kinds of layers:

* **Convolutional layers** create a feature map to predict the class probabilities for each feature by applying a filter that scans the whole image, a few pixels at a time.
* **Pooling layer** scales down the amount of information the convolutional layer generates for each feature and maintains the essential information.
* **Fully connected input layers** “flatten” the outputs generated by previous layers to turn them into a single vector that can be used as an input for the next layer.
* **Fully connected layers** apply weights over the input generated by the feature analysis to predict an accurate label.
* **Fully connected output layer** generates the final probabilities to determine a class for the image.

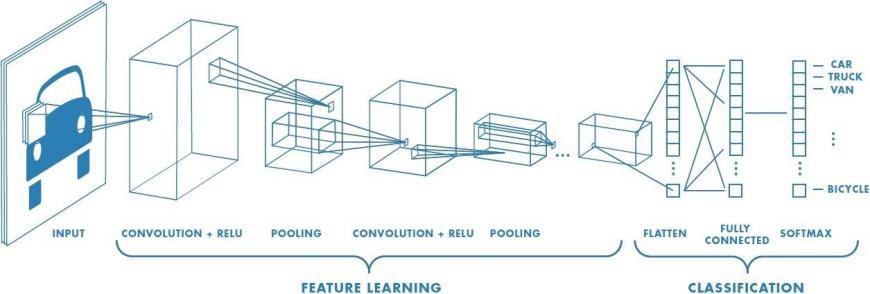


Figure 3-4 CNN architecture

The architecture of a CNN is the way in which the layers are structured and their order and that is a key factor in determining its performance and efficiency. which elements are used in each layer and how they are designed will often affect the speed and accuracy with which it can perform various tasks.

## 3.4. Comparative Study of Previous Work

[1] proposed a system that analyses football using domain knowledge. The process was dependent on motion detection and object recognition.

[2] presents techniques to detect and extract the highlights by analyzing the video contents.

[3] proposed a novel object detecting and tracking method in order to detect and track objects necessary to describe contents of a football game.

[4] presented an efficient framework for analysis and summarization of football videos using cinematic and object-based features

[5] presented a framework for shot classification in sports video.

## 

## 

## 

## 

## 

## 

## 

## 

## 3.5. Implemented Approach

The Proposed system is presented to avoid most of the problems discussed in the previous sections, these solutions are involved in each of the stages of the proposed system. In the preprocessing stage, reading all frames of a video is clearly an overhead on the system as most football matches are two hours long and also the differences between two consecutive frames doesn’t carry much information because very little has changed between the frames so frame skipping is done to ensure that there is no redundancy in the consecutive frames and also make the system much faster.

Shot boundary detection’s purpose is segmenting the input video into small video shots by applying image processing and histogram comparison techniques. In the shot classification phase, the system applies different algorithms, namely; Grass ratio, Face Detection, Deep learning and image processing techniques to classify the shot into one of the following classes: wide – medium- close – close out – logo. The logo classification is used for replay detection which is an important factor in determining which of the shots are important and which are not.

In The event detection phase each shot is considered separately and compared with the previous shot to determine if a goal has happened between the two shots by comparing the scoreboard in the upper left corner. Two methods are proposed to detect a goal event, Optical character recognition (OCR) and structural similarity image index (SSIM).then, analyzing audio to detect where are the spikes of volume which is probably an important event in the match and deserves further analysis, then detecting if a goal mouth appears in the shot or not. all those factors will come in play later in the summarization phase

Lastly, the summarization phase which takes into account the output of the previous phases to decide if the shot is important and should be included in the final video or should be discarded. For example, if a shot contains a goal then it is important or if a shot has a goal mouth and the audio level is high then it is important or it’s a replay shot and a replay of course is for something important. Then combining the important shots only in a video and that is the output of the system

# 

# 

# Chapter 4: System Design and Architecture

In this chapter, we present the architecture of the proposed system and discuss each phase of the five phases that the system comprises in detail.

## 4.1. Overview and Assumptions

The proposed system comprises five phases; in the pre-processing phase, the system reads the video stream into patches of frames. Then, in the shot Boundary phase is applied on every patch of frames to segment the video into separate shots i.e. finding the transition from a camera into another. Then each shot is processed separately in the following three phases, the shot classification phase in concerned with classifying each shot for example a wide shot is when the whole pitch is visible, A close shot is when the players faces are visible etc. then the event detection phase in which each shot is analyzed to determine if a goal happens in it or not or if it the audio level in this shot is high with respect to the whole video or if the goal mouth appears in the shot . And lastly, the summarization phase in which the output of all previous phases is analyzed to determine which shot is important and should include in the final video and which shots to discard.

Due to lack of datasets of matches and copyright laws this project is a proof of a concept which mainly works on the English premier league but that does not mean that it does not work on other leagues it just means that for laخsly to work on other leagues it just needs matches to be able to detect the logo and the position of the scoreboard which are important factors in our system

## 

## 

## 

## 

## 

## 

## 4.2. System Architecture

The preprocessing stage, reading all frames of a video is clearly an overhead on the system as most football matches are two hours long and also the differences between two consecutive frames doesn’t carry much information because very little has changed between the frames so frame skipping is done to ensure that there is no redundancy in the consecutive frames and also make the system much faster.

Shot boundary detection phase is proposed to segment the whole input video stream into small video shots by applying a combination of image processing techniques and histogram comparison techniques. In the shot classification phase, the system applies different algorithms, namely; Grass ratio, Face Detection, Deep learning and image processing techniques to classify the shot into one of the following classes: wide – medium - close – close out – logo. The logo classification is used for replay detection which is an important factor in determining which of the shots are important and which are not.

In The event detection phase each shot is considered separately and compared with the previous shot to determine if a goal has happened between the two shots by comparing the scoreboard in the upper left corner. Two methods are proposed to detect a goal event, Optical character recognition (OCR) and structural similarity image index (SSIM).then, analyzing audio to detect where are the spikes of volume which is probably an important event in the match and deserves further analysis, then detecting if a goal mouth appears in the shot or not. all those factors will come in play later in the summarization phase

Lastly, the summarization phase which takes into account the output of the previous phases to decide if the shot is important and should be included in the final video or should be discarded. For example, if a shot contains a goal then it is important or if a shot has a goal mouth and the audio level is high then it is important or it’s a replay shot and a replay of course is for something important. Then combining the important shots only in a video and that is the output of the system

### 4.2.1. Block Diagram

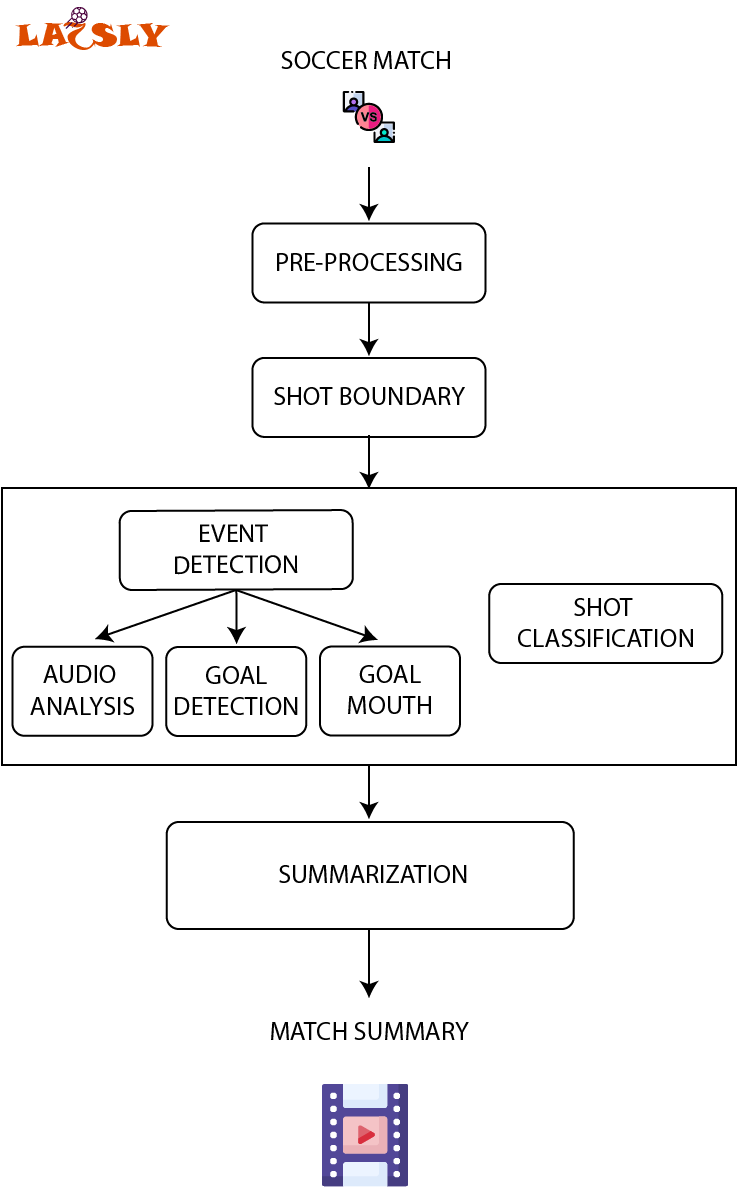


Figure 4-1 LAخSLY Block Diagram

## 4.3. Pre-processing phase

### 4.3.1. Functional Description

As the first phase of the system its functions are simple which are to read the video to be summarized form storage, skip unnecessary and redundant frames to avoid any unnecessary calculations and store the frames in a proper data structure for subsequent phases to process.

### 4.3.2. Modular Decomposition

As mentioned before, reading all frames of a video is clearly an overhead on the system as most football matches are two hours long and also the differences between two consecutive frames doesn’t carry much information because very little has changed between the frames so frame skipping is done to ensure that there is no redundancy in the consecutive frames and also make the system much faster.

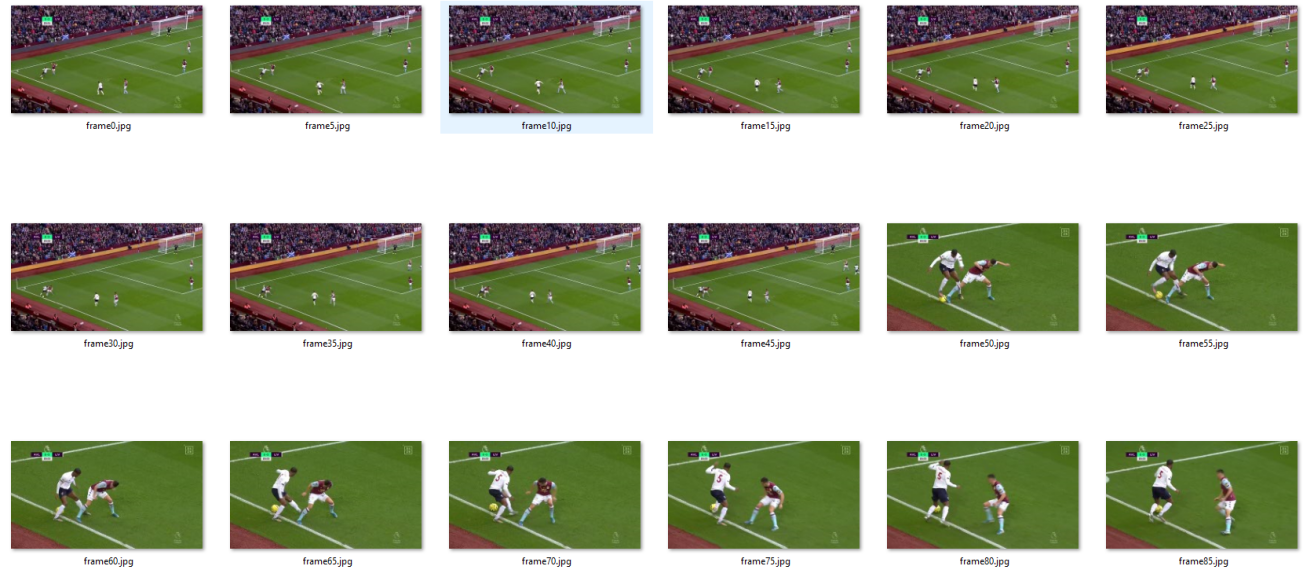


Figure 4-2 Frames of video

The frame skipping process consists of reading a frame then ignoring k subsequent frames then appending the k+1 frame. The value of k is best within the range (5,10) since less than 5 there will be useless frames and larger than 10 there will be missing important information.

### 

### 

### 4.3.3. Design Constraints

Reading a 2-hour video into a computer’s memory especially if the video is of higher quality would require too much resources, the solution for this is to divide the input frames into patches each is roughly 2000 frames which is good for an eight-gigabyte memory. The processing of the patch through all phases of the system except the summarization phase then repeating the pre-process phase to obtain the next patch and that whole process continues until the end of the video then the summarization phase can start.

## 4.4. Shot Boundary

As mentioned in the background section, a transition between two cameras is a transition between two different shots. In this section we discuss how to detect those transitions to divide the video into shots and analyses these shots individually.

### 4.4.1. Functional Description

We have two types of transitions in a football match

1. Instant (cut) transition
2. Logo transition.

A logo transition is When there is an editing effect happens in which the logo of the competition swipes over the screen to transition from a live feed to a replay of something that has already happened in the match and vice versa.



Figure 4-3 example of an instant cut



Figure 4-4 example of logo transition

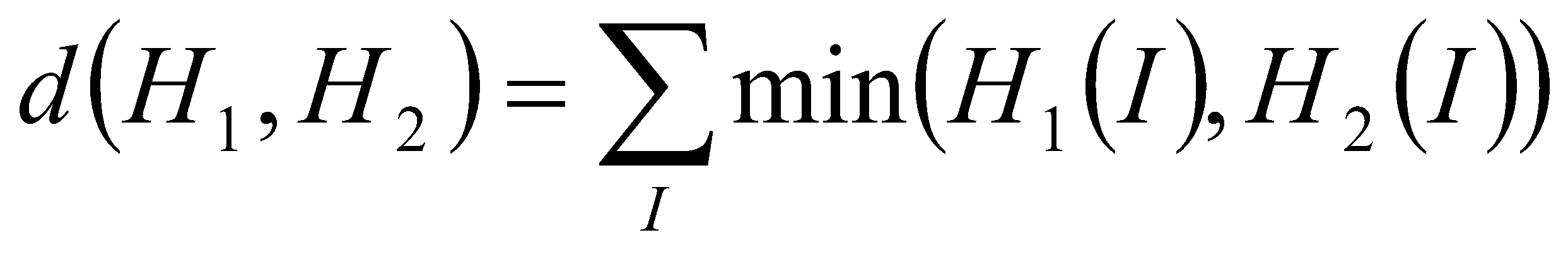
### 

### 4.4.2. Modular Decomposition

We use color histogram of two consecutive frames as the main identifier for the shot boundary detection phase since color histograms are robust to moderate object and camera motions, we represent a frame by its color histogram, which is defined in the RGB space i.e. each bin in the histogram corresponds to a combination of values (r,g,b). we used 64 bins for each channel which means that now we get a 64x64x64 array, where each element corresponds to a number of pixels with a particular (r,g,b) value. For example, histogram [50,100,150] denotes the number of pixels who have the values B=50, G=100, R=150.

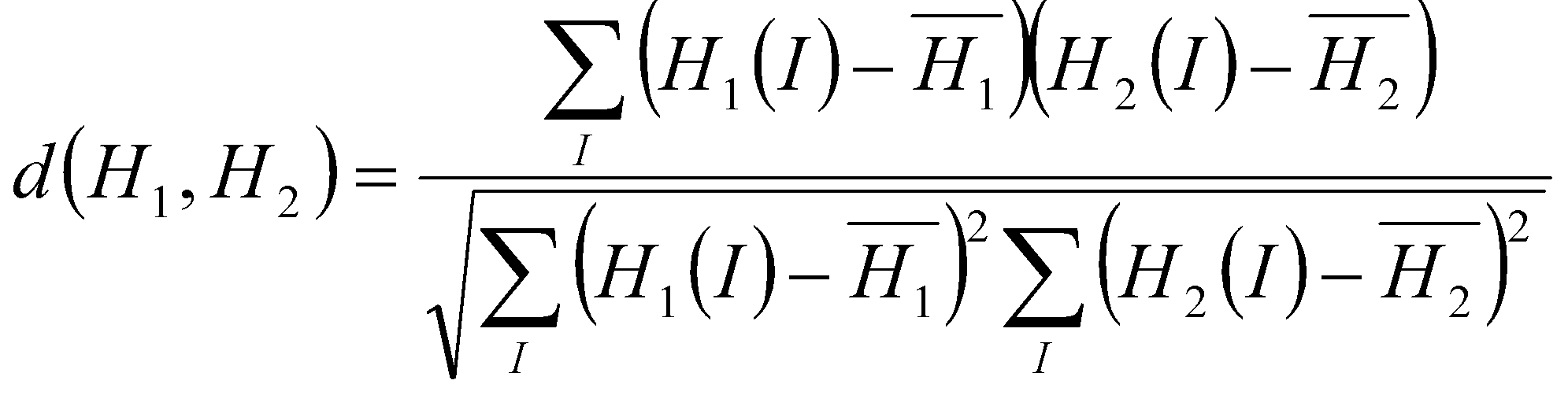
Comparing histograms is a powerful and easy process to know if two images are different or not and measure the similarity. There are many methods to compare two histograms but the ones considered are discussed below [6].

- Histogram Intersection of two histograms H1 and H2.

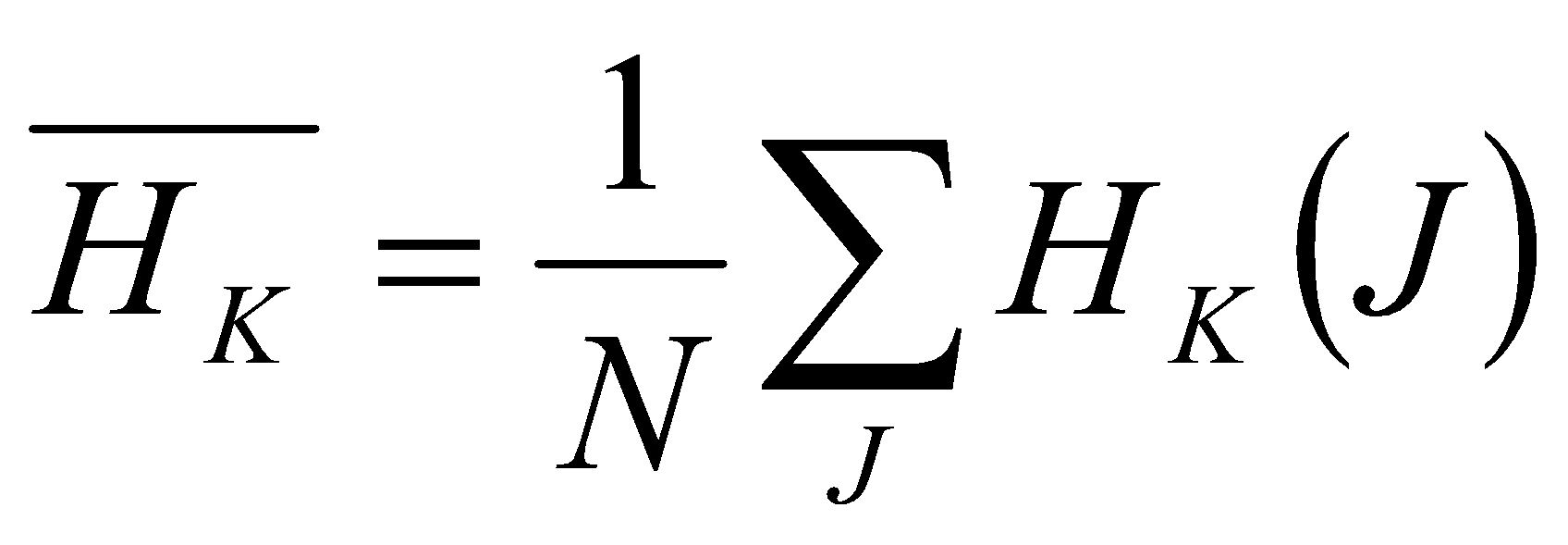


This method simply compares, for each bin, the two values in each histogram, and keeps the minimum one. The similarity measure is then simply the sum of these minimum values. Consequently, two images having histograms with no colors in common would get an intersection value of 0, while two identical histograms would get a value equal to the total number of pixels.

- Histogram Correlation of two histograms H1 and H2.



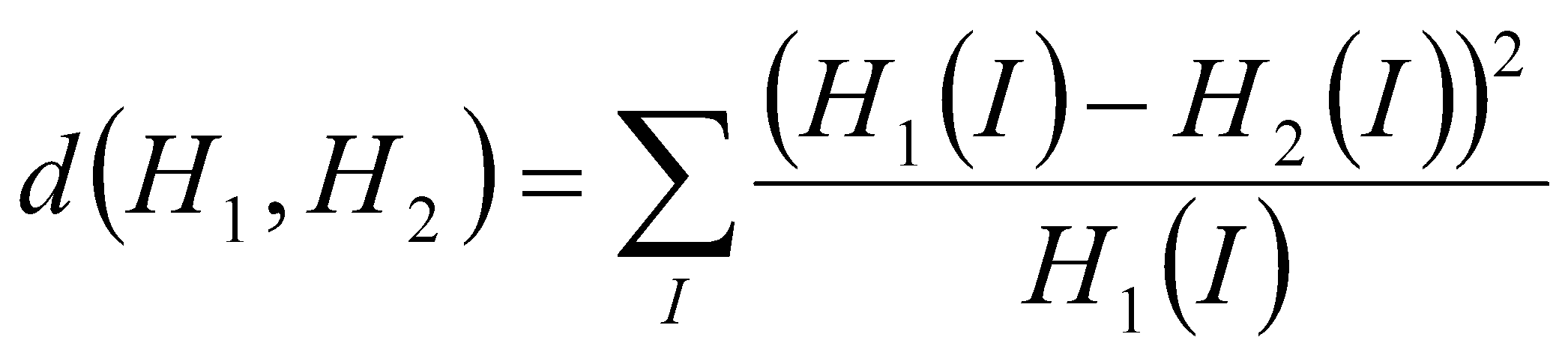
where



is based on the normalized cross-correlation operator used in signal processing to measure the similarity between two signals, cross-correlation is a [measure of similarity](https://en.wikipedia.org/wiki/Similarity_measure) of two series as a function of the displacement of one relative to the other. This is also known as a sliding [dot product](https://en.wikipedia.org/wiki/Dot_product) or sliding inner-product.

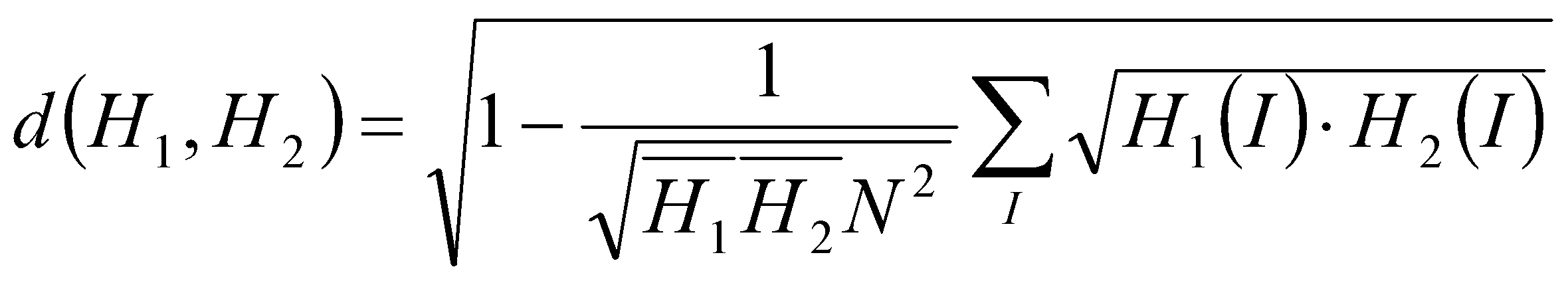
For the *Correlation* and *Intersection* methods, the higher the metric, the more accurate the match

- Chi-Square



It’s a simple method sums the normalized square difference between the bins of the histogram

- Bhattacharyya distance



Bhattacharyya distance is used in statistics to estimate the similarity between two probabilistic distributions.

For Chi-Square and Bhattacharyya distance methods, the higher the metric, the lower the match between the two histograms

Through trial and error, it was shown that the intersection and correlation methods are the most accurate therefore reliable as they produced consistent outcomes in the instant cut as well as the gradual cut on the other hand Chi-Square and Bhattacharyya distance had low accuracy in classifying the logo transitions. So, the proposed algorithm is dependent on the intersection and correlation methods.

The detection of logo transitions in sports video is particularly difficult because of the high color correlation between two shots so Instead of computing the histograms for every consecutive frame pair, the comparisons are performed between frame i and frame i + k where k is an offset as demonstrated before in the pre-processing phase.

The two frames below are obviously consecutive frames from the same shot i.e. there is no cut between them. That should be reflected in the values of histogram intersection and correlation.



Figure 4-5 example of consecutive frames from the same shot

*Histogram Intersection = 9.04120311407678*

*Histogram correlation = 9.971235345958373*

The values are higher than the thresholds mentioned in the above algorithm (the thresholds are based on trial and error) so no cut.

Here is another example to fully explain how the shot boundary algorithm works.

These are also consecutive frames but there is a transition (cut) between them



Figure 4-6 example of consecutive frames with a transition(cut) between them

*Histogram intersection = 0.24404063194742776*

*Histogram correlation = 0.002219072222348816*

Both values are smaller than the thresholds so we perform step 4 of the algorithm in which dividing the two frames into blocks, each block 150px x 150px and calculate intersection and correlation between individual blocks then, according to the values of intersection and correlation of the blocks to consider whether that block is considered full changed or half changed or not changed at all.

Then, step 9 in which we count how many blocks have changed and calculate the percentage with respect to all blocks. In the above two frames the percentage of changed blocks is 100% i.e. all frame blocks have changed so, there is definitely a cut.

Another example to fully demonstrate the importance of having two methods.

The following two frames are consecutive but in the midst of a logo transition.



Figure 4-7 example of two frames are consecutive but in the midst of a logo transition

*Histogram intersection = 2.1293090697099615*

*Histogram correlation = 9.844120057485238*

Here the two methods disagree, the intersection indicates that there are minor similarities in the two frames but the correlation indicates that they are variations of each other i.e. the histogram is a shifted version of the other histogram for example.

So, we resort to the blocks process once again to determine if there is a cut or not. The percentage of changed blocks is 37% which makes sense because there are areas that didn’t change but according to the threshold this is still considered a cut and that is logical because it’s a logo transition.

**Algorithm:**

1. Get color histograms of the two frames in RGB space.
2. Calculate histogram intersection and correlation between the two histograms
3. If intersection >6 and correlation >5 then no cut.
4. Divide the two frames into blocks, each block 150px x 150px and calculate intersection and correlation between individual blocks.
5. If intersection <4 and correlation <4 then block is 100% changed.
6. If intersection >4 and correlation <4 then block is 75% changed.
7. If intersection <4 and correlation >4 then block is 25% changed.
8. If none of the above conditions are met then the block is not changed.
9. Count changed blocks and calculate percentage of changed blocks.
10. If the percentage of changed blocks > 30% then Cut else no cut.

**Flow Chart**

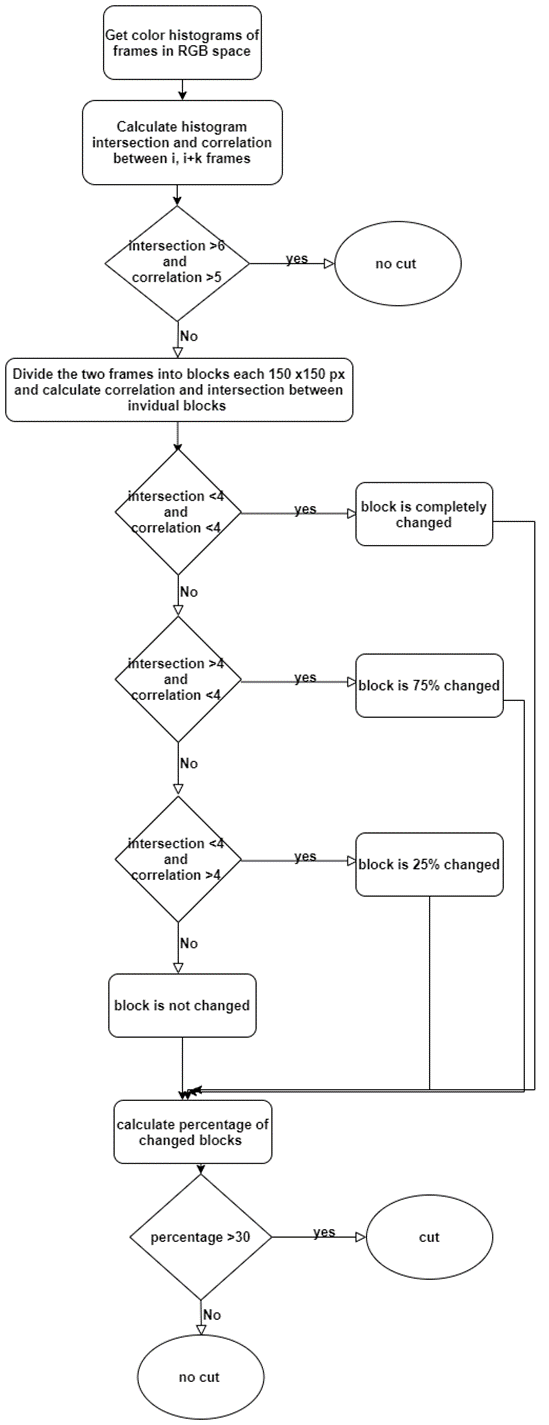


Figure 4-8 Flow Chart of Shot Boundary

### 4.4.3. Design Constraints

Finding the shot boundary in a video is a vast and growing field of research that recently attracted many researchers. one of the main hurdles that is in the way of researchers is the fade transition in which the last frames of a shot fade (dissolve) into the next cut. This transition is so much harder to detect compared to a hard cut or a logo transition. our algorithm’s performance in detecting such a cut is mediocre but fortunately a fade cut rarely happens in a football match

## 4.5. Excitement Event detection

Exciting and important events in football matches happen with some indicators that can be used to identify them, high audience and commentator voice is a good indicator of an important event in the match, and the higher the voice, the more important the event. a change in the scoreboard numbers is an indicator of scoring a goal. most of the important events during the match tend to happen near the goal mouth, which is an indicator of a goal or attack.

In the following sections we will make use of these indicators starting by the audio processing section that will be used to determine the shots that contain high audio levels, then in the goal mouth detection section we will try to find out whether a shot contain a goal mouth or not and finally, in the goal detection section we will try to detect the change in the scoreboard to find out whether a goal is scored.

### 4.5.1. Audio Processing

Intensity, muteness and pitch of the sound of the commentators and audience are very effective for measuring excitation. The intensity of the sound volume is the simplest feature of the audio that can be used as an indication, so in this approach we get the periods of time in the video where the intensity of the volume is high which can indicate that an interesting event happened at this period.

The algorithm below samples an audio level that is the average value from every 10 seconds in the match then calculates the difference between consecutive averages i.e. consecutive 10 seconds of the match.

If the outcome of the differencing operation is positive +ve that means the volume is increasing so if multiple increasing instances are consecutive, that is an indication of a period in the match where the audio levels are rising.

We find that by calculating the 90% of the values in the averaged volumes.

for example: averaged volumes = [1,2,3,4,5,6,7,8,9,10, 11,97,98,99,100]

so, having the averaged array, we take the largest 10% of it

so, values > 90% are [91, 92,99,100] i.e. the larger 10% of the array.

**Algorithm:**

1. Read a video clip.
2. Extract audio from the video clip.
3. Get the average volume of each 10 seconds.
4. Get the difference between every two averages then detect the increases and decreases in volume.
5. Determine peaks indices of volumes.
6. Get peak volumes.
7. Get Times of peaks having volume level > 90%.

**Flow Chart**

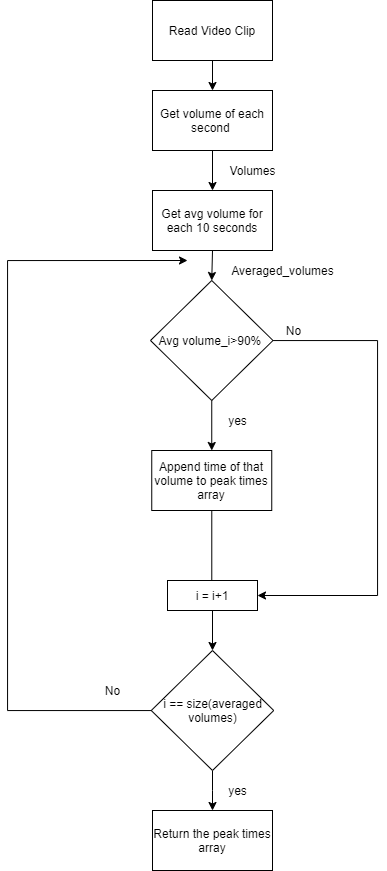


Figure 4-9 Flow chart of the audio processing

### 

### 4.5.2. Goal-Mouth Detection

In football, the goal-mouth appearance can indicate that an interesting event is happening such as a goal, penalty, direct free kick or shooting towards the goal, etc. So detecting the goal-mouth can help in determining which even to be selected in the final output by differential between those interesting events and other not interesting events such as dull passes, or other shots of audiences or coaches, etc.



Figure 4-10 Examples of important events containing goal post

Goal-mouth is important when it appears in wide shots and medium shots. For each one there are different image processing approaches to detect if the shot contains goal-mouth or not due to different camera angle, point of view, etc.

For wide shots, the appearance of two or three parallel field lines in a wide view can be used to indicate the occurrence of the goal-mouth. The appearance of goal-mouth and parallel field lines are highly correlated. This information of parallel lines which indicates the penalty box is very useful for goal-mouth detection. The information of the parallel lines is more reliable than the information of the goalpost here in the wide shot, since the goalposts detection may fail due to the cluttered background pixels and the environment interference. Here, we use a canny edge detector with minimum value and maximum value threshold of 60 and 120 and Hough Transform to detect the lines of minimum length of 80 giving the endpoints of each line. The parallel lines are detected by checking if the difference between their angles is less than a tolerance threshold, and their slope angles range from 140 to 170 or range from 10 to 40. When parallel lines tilt to left, it implies a right goal, otherwise a left goal [7].

For medium shots, RGB image is converted into Gray color image in order to do further image processing operations. Noise in the image has been reduced using Gaussian low pass filter. Canny edge detection algorithm with minimum value threshold of 50 and maximum value threshold of 100 is used to detect the edges in the image for identifying the goal post region. The goal post region is obtained by applying masking on the edge detected image, the masking is derived by mapping the edge detected image with the grass green color image. The horizontal and vertical lines parallel lines of minimum length of 60 are identified to find the goal-mouth in the region of interest using Hough transform [8].

**Algorithm*:***

1. Take the input image frame and its type (wide or medium).
2. Convert RGB image to Gray color.
3. Remove noise using a Gaussian filter.

For wide type:

1. Get the edges using canny edge detection with maxVal and minVal equal to 60 and 120 respectively.
2. Obtain the lines endpoints of minimum length 80 from the edge detected image using Hough transform.
3. If there are two or more lines parallel with angles in range 140 to 170 or 10 to 40 then return true and goal-mouth is detected.

For medium type:

1. Get the edges using canny edge detection with maxVal and minVal equal to 50 and 100 respectively.
2. Convert frame image to HSI color space to obtain mask by segmentation of grass green color.
3. Apply mask to edge detected image.
4. Obtain the lines endpoints of minimum length 60 from the edge detected image using Hough transform.
5. If there are two or more lines parallel then return true and goal-mouth is detected.

**Flow Chart**

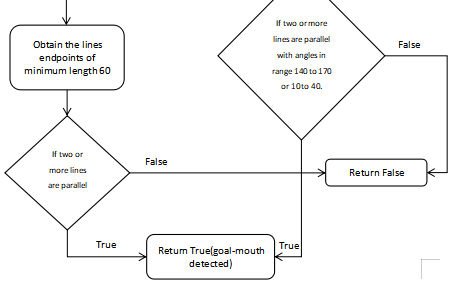
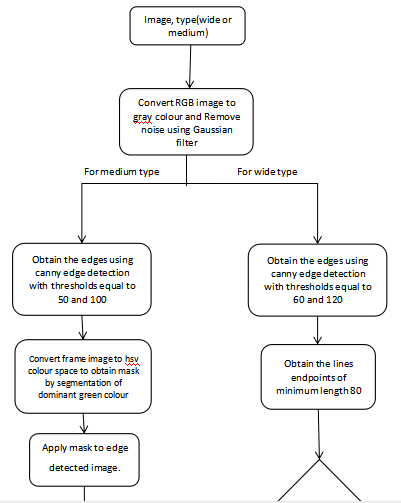


Figure 4-11 Flow Chart of goal-mouth

### 

### 4.5.3. Goal detection

A goal is arguably the most important event is the sport of football and subsequently if a goal happens in a match it must be included in the summary of that match. so goal detection is of great importance. During the match, the score is displayed in the scoreboard.

The scoreboard is a caption region (usually at the top left) distinguished from the surrounding region, which provides the information about the score of the game and the match time. The score board dimensions and position is always constant in a certain league so detection of the scoreboard itself is not needed.



Figure 4-12 Examples of Scoreboards

When a goal is scored the scoreboard will change to indicate the new score, so detecting that change is the simplest and best approach to tackle such a problem. discussed below are two approaches to detect the change in the scoreboard during the match.

The scoreboard score region can be segmented into 3 parts (number, separator, number). The region is segmented using the vertical histogram. The values with 0 histogram are the separators, we discard them and get the numbers.

#### 4.5.3.1. Goal detection using OCR

In this approach, we store the score of the match which is initially 0-0 then we apply the OCR on the number regions of the scoreboard to extract the number string. After this we compare the results with global variables which stored the current score.



Figure 4-13 Example Before and after a goal

extracting the number did not require implementing a method from scratch as there are plenty of open source OCR libraries available and it is out of scope for this project to implement OCR. In this approach we used Tesseract engine for OCR, it is free software, developed by Google.

**Flow Chart**

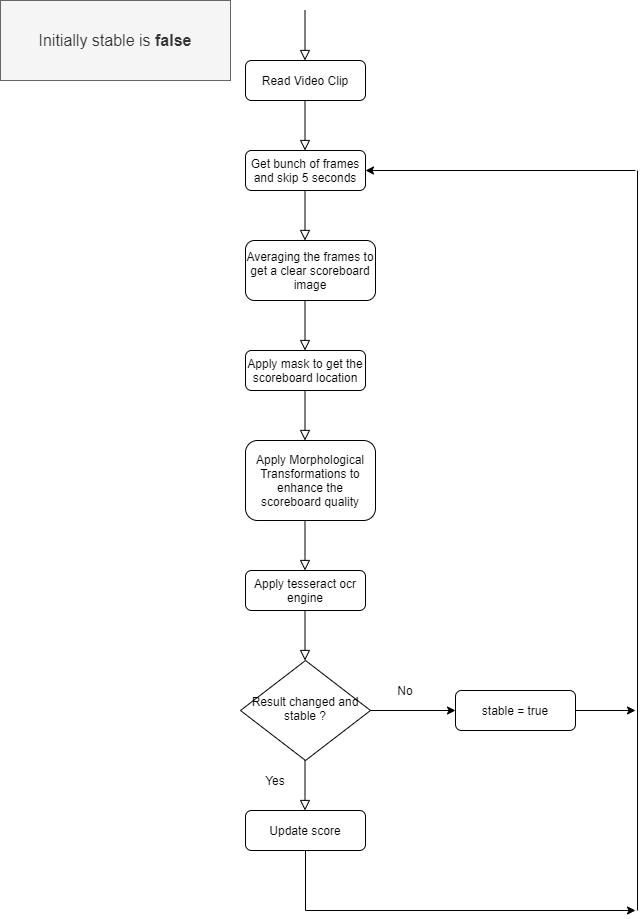


Figure 4-14 Flow Chart of Goal detection using OCR

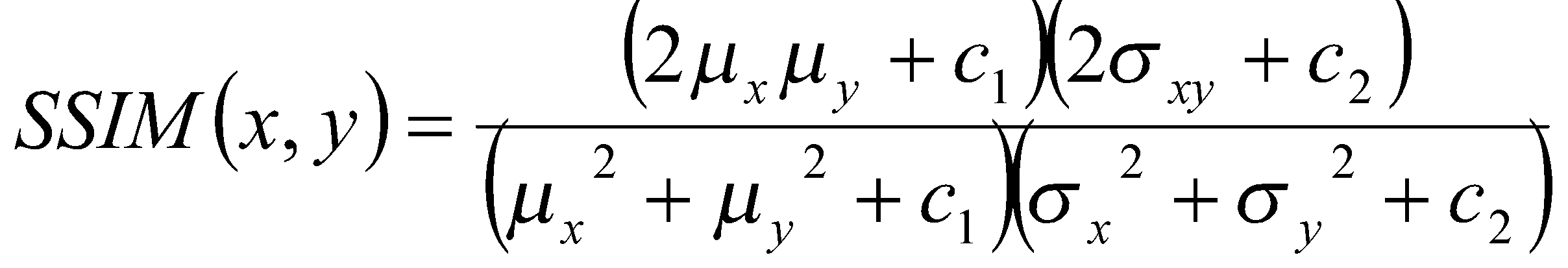
#### 4.5.3.2. Goal Detection with structural similarity image index (SSIM)

In this approach, changes in the scoreboard are detected with the structural similarity image index (SSIM), The SSIM method is used for predicting the perceived quality of digital television and cinematic pictures, as well as other kinds of digital images and videos. and used for measuring the similarity between two images.

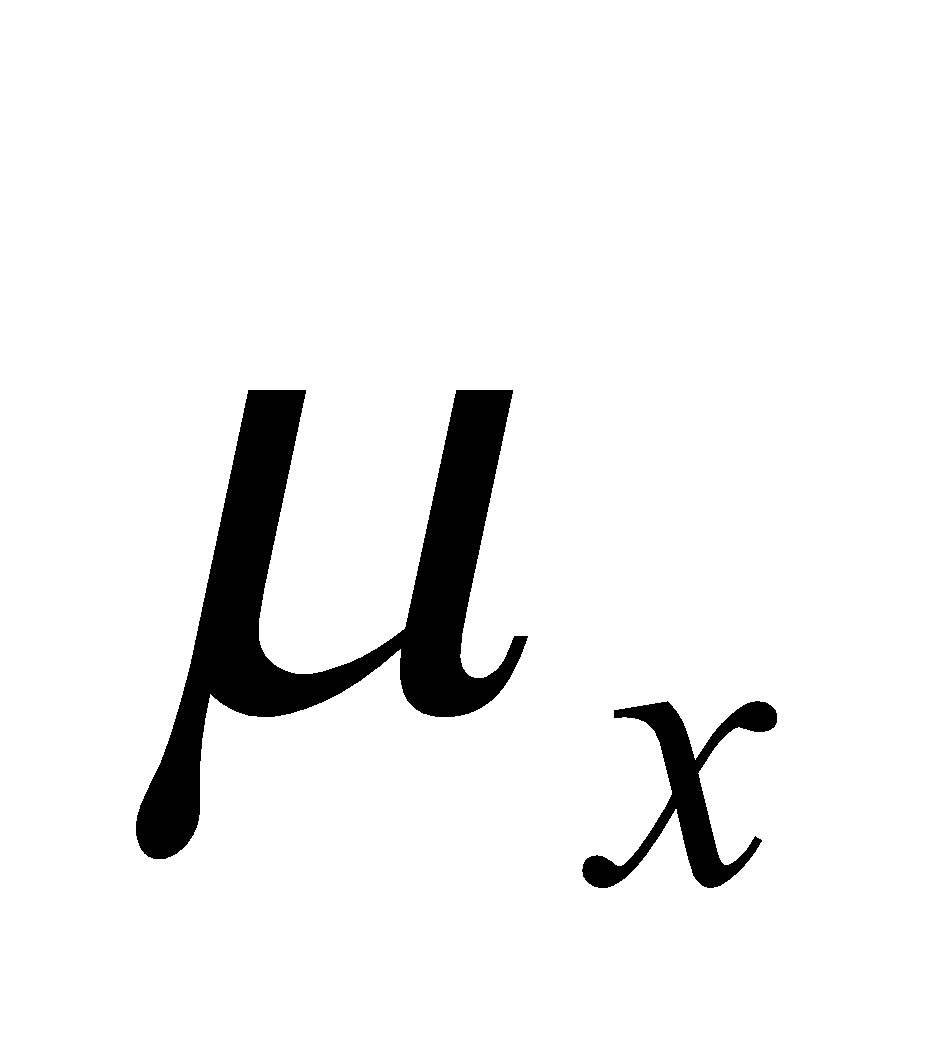
The difference with respect to other comparing techniques such as Mean square Error (MSE) or Peak signal-to-noise ratio (PSNR) is that these techniques estimate absolute errors; on the other hand, SSIM is a perception-based model that considers image degradation as perceived change in structural information. while also incorporating important perceptual phenomena, including both luminance masking and contrast masking terms. Structural information is the idea that the pixels have strong inter-dependencies especially when they are spatially close. Luminance masking is a phenomenon whereby image distortions (in this context) tend to be less visible in bright regions, while contrast masking is a phenomenon whereby distortions become less visible where there is significant activity or "texture" in the image [9].

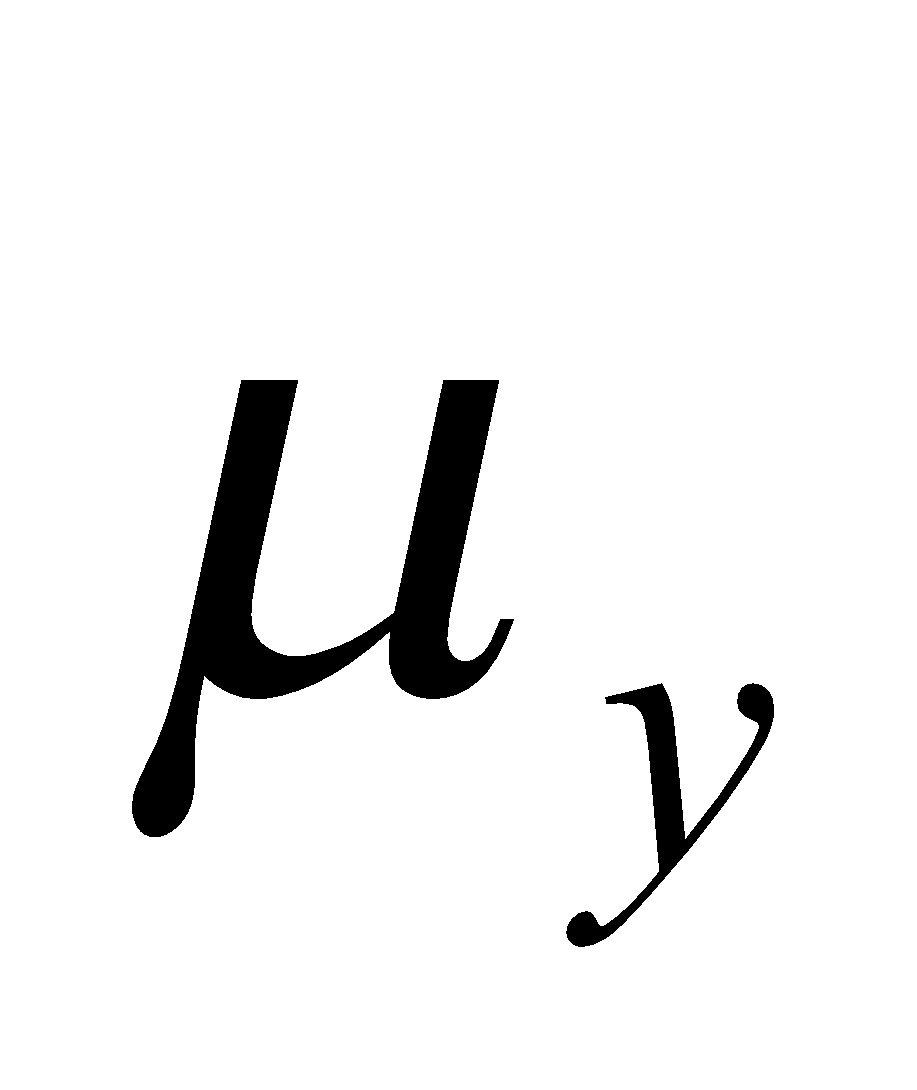
It is an enhanced way to detect if the image changed or not. It uses the difference between the scoreboard and get the mean, variance and illuminance between the two scoreboards and get if the scoreboard changes or not. If the scoreboard changes then it is an event (goal or substitution).

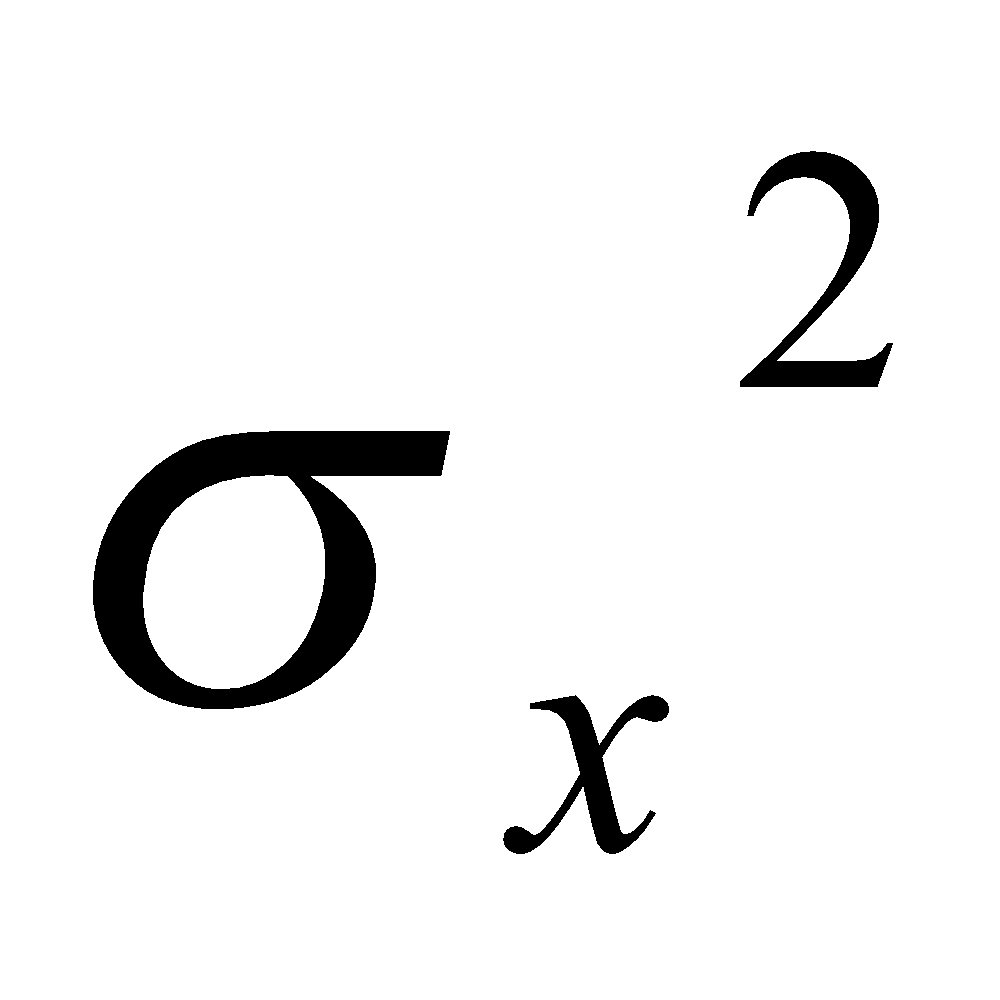
The SSIM index is calculated on various windows of an image. The measure between two windows x and y:

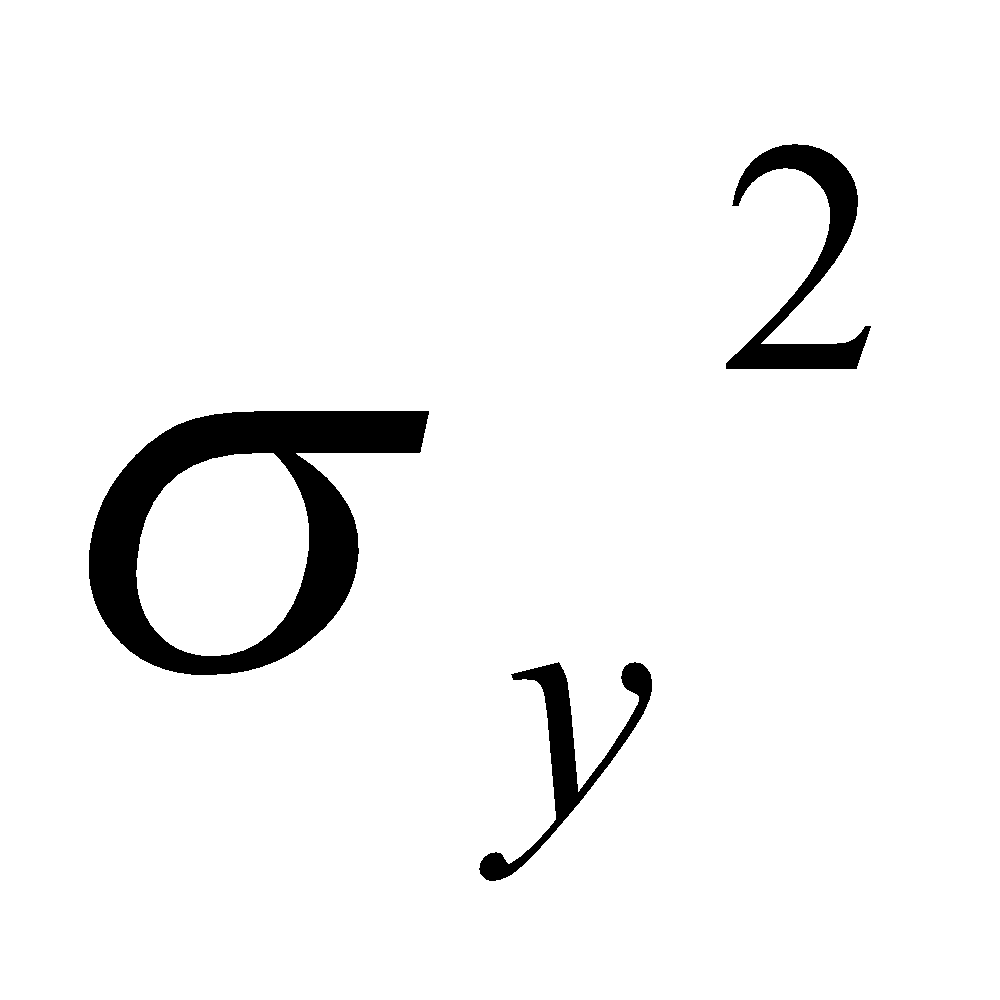


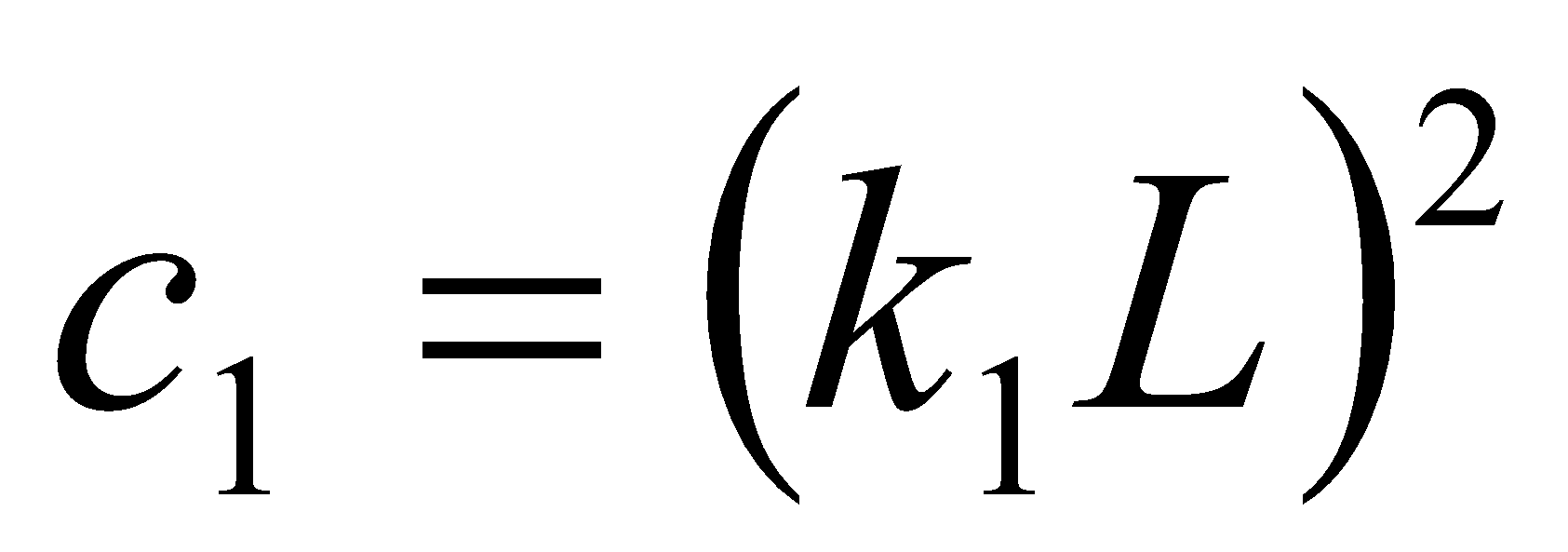
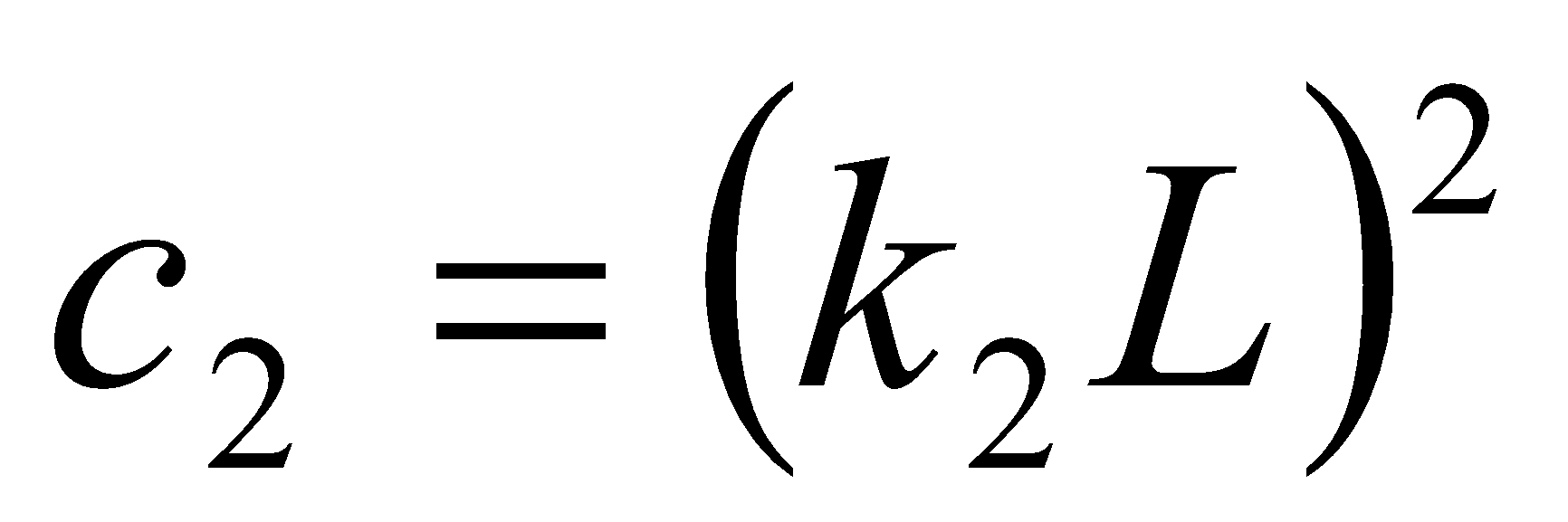
where

 is the average of x

 is the average of y

 is the variance of x

 is the variance of y

 and 

with L = is the [ratio](https://en.wikipedia.org/wiki/Ratio) between the largest and smallest values that a certain quantity can assume

and k1= 0.01, k2 = 0.03 by default

**Flow Chart**

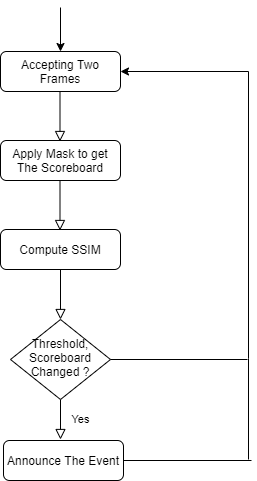


Figure 4-15 Flow Chart of Goal detection using SSIM

After trying with different matches with different scores we determined that the threshold for change is best to be around 70%.

The OCR approach has an advantage over the SSIM method which is that it returns only the goals meaning we see the results as strings, so comparing is easy. But for the SSIM method, the results are percentages of change so it’s threshold dependent and results for both are discussed in chapter 6.

## 4.6. shot classification

The purpose of this phase is to classify the shots detected by the shot boundary phase. We can classify a sports video shot into one of four categories wide, medium, close-up and close out (out-of-field).

1. **Wide Shot:** A wide shot displays a wide view of the field; a wide shot must shot must contain a large part of the green pitch
2. **Medium shot:** A medium shot is where the whole body of the player is visible, medium shot is a zoomed-in view of a particular region in the field.
3. **Close:** A close shot is when the player or the referee’s face appears up close.
4. **close-out (Out-of-field) Shot:** The audience, coach, and other Out-of-field shots.
5. **logo**: A logo shot is when the logo of the competition swipes the screen to transition from the live feed to a replay and vice versa

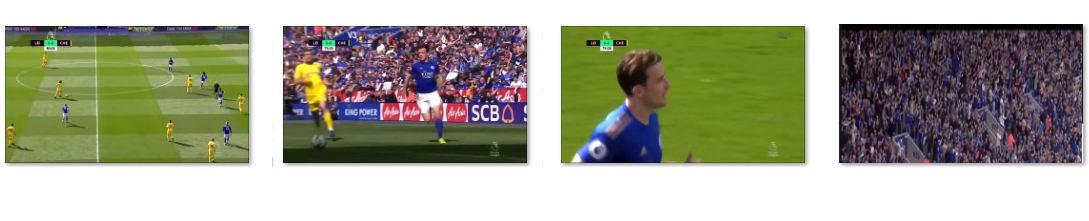
**

Figure 4-16 Types of shots

This classification will help us in the summarization phase as most important events happen in the wide shots of the match then the director cuts to a medium shot or close on the player involved in that important event then replay of the important event and that involves a logo transition. Detecting that order would be the backbone of the summarization phase later on.

In the following sections we discuss four approaches to classification of shots:

1. Shot classification using Grass Ratio.
2. Shot Classification using Face Detection.
3. Shot classification using Deep learning.
4. Shot classification using image processing and Machine Learning.

### 

### 

### 

### 4.6.1. Shot classification using Grass color ratio

The playing field usually has a distinct tone of green that may vary from one stadium to another, even in the same stadium’s green color may change due to weather and lighting conditions [4].



Figure 4-17 Different tones of green color

We present an algorithm based on HSI color space which is more robust to change in lighting conditions than RGB

**Algorithm for Grass ratio:**

*1: read the input video.*

*2: for each frame do.*

*2.1: convert frame from RGB to HSI.*

*3: define the color range that covers the different variations of the play field’s green color.*

*4: for each pixel do.*

*4.1: if the HSI components range as follows*

*4.1.1: 0.4 > Hue > 0.15*

*4.1.2: 0.6 > Intensity > 0.2*

*4.1.3: 1 > Saturation > 0.1*

*4.2: set the pixel to ‘White’*

*4.3: otherwise, set the pixel to ‘Black’*

*4.4: end if*

*5: end for*

*6: end for.*

The above numbers have been developed based on try and error on different football matches from the English premier league with different weather and lighting conditions.

The grass color algorithm yielded good results as illustrated in figure:



Figure 4-18 example of grass color algorithm results

The next step is to use the grass ratio (G) obtained from above to classify the shot as one of the following classes:

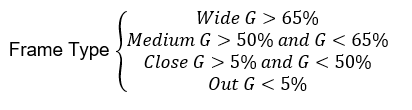
1. wide
2. medium
3. close
4. out of filed

The proposed algorithm is based on a specific threshold (range) for grass ratio (G); which was developed by observing different matches in many lighting and weather conditions to appropriately define a range that will cover the four shot types.

Classification of a shot into one of the discussed three classes is based on spatial features. Therefore, shot class can be determined from a single key frame or from a set of frames selected according to a certain criterion.

For example: if the frame is in a wide shot that means that the whole field is visible which means that G would be more likely ~70% of the frame

Thinking intuitively, G would be almost zero for out-of-field frames, a low G value in a frame corresponds to a close, while high G value indicates that the frame is a long view, and in between, a medium view is selected.



***For each shot***

1. *Choose a set of keyframes in the shot.*
2. *Compute grass ratio G of these frames.*
3. *Classify frames.*
4. *Determine the majority type of the frames and assign it to the shot.*

The set of the key frames is based on the length of the shot, if the shot is small (less than 20 frames) then the whole shot is included and if the shot is larger than 20 frames then a of sample of 20 frames are sampled uniformly from the shot from start to end.

Choosing the set of frames also depends on the computation power running the code.

Due to the computational simplicity of the proposed algorithm, computing the grass ratio of all frames in the shot would not be a big overhead.

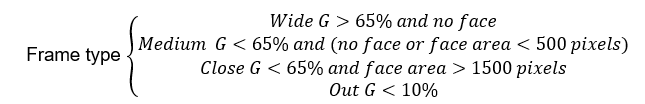
Although the accuracy of the above simple algorithm is sufficient but it does not meet the desired outcome. The accuracy of the above algorithm is roughly 60% (i.e. it classifies 60% of the shots correctly)

### 

### 4.6.2. Shot classification using Face detection

In the observation phase mentioned above it can be said that the Wide shot does not contain any clear faces (large enough to be recognized by a human or a computer), in the medium shot a face can or cannot be recognized depending on the angle of the camera filming the shot, in the close shot a face can be clearly recognized as it covers a large area of the frame.

With that said, combining the face detection model and the Ratio G to obtain better results in classification.



**Algorithm:**

***For each shot***

1. *Choose a set of keyframes in the shot.*
2. *Compute grass ratio G of these frames.*
3. *Determine whether the frames contain faces or not and if it does get the Area of*

*of the bounding box*

1. *Classify frames.*
2. *Determine the majority type of the frames and assign it to the shot.*

The obtained results are better than using grass ratio only but the computational time and complexity is much worse because the face detection model is very complex and time consuming.

This approach yielded good results (~70%) but we didn’t spend much time refining it because the face detection model is pre trained and we wanted our classification method to be implemented from scratch

### 

### 

### 

### 4.6.3. Shot Classification using deep learning

This approach requires creating a huge dataset of Frames that covers all the needed categories of (wide - medium - close - close-out - logo) and there is no available dataset set that satisfies these requirements so we had to resort to manual labor and create the dataset from scratch.

The Frames are Extracted from Matches and assigned labels manually according to their respective category to construct the data set, for example, we extract the frames from the match then separate the wide frames from the close from the medium etc. It’s an exhaustive process to create labeled data but that was the only option to be able to train a deep learning model. our data consists of 8,000 images per each class (Total 40,000) all labeled with their appropriate respective class. Now we move on to the CNN model.

we didn’t invent the wheel in choosing the architecture of the CNN model (to understand what is the architecture of a CNN and what is a CNN model in general refer to chapter 3 section 4:Background on Convolutional Neural Networks) we chose a famous architecture that has been used for years and proven to be efficient in classifying images.

The LeNet architecture was first introduced in [10], Gradient-Based Learning Applied to Document Recognition. As the name of the paper suggests, the authors’ implementation of LeNet was used primarily for OCR and character recognition in documents.

The LeNet architecture consists of the following layers:

**INPUT > CONV > RELU > POOL > CONV > RELU > POOL > FC > RELU > FC**

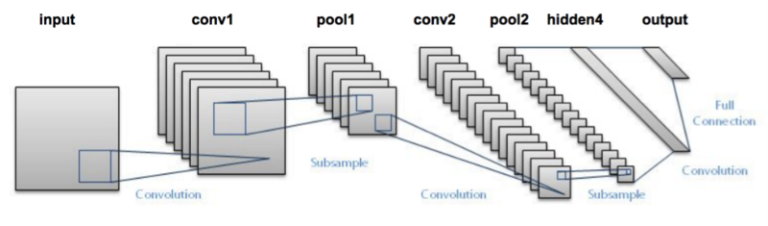
****

Figure 4-19 LeNet architecture

After 20 epochs the accuracy reached 86% which is much better than the previous techniques.

### 4.6.4. Shot Classification using image processing and machine learning

[10] proposed a compute-easy and efficient algorithm for classifying the frames with a high *G* value. We define regions by using the Golden *Section* spatial composition rule which suggests dividing up the frame in 3:5:3 proportion in both directions as shown in figure 4-20

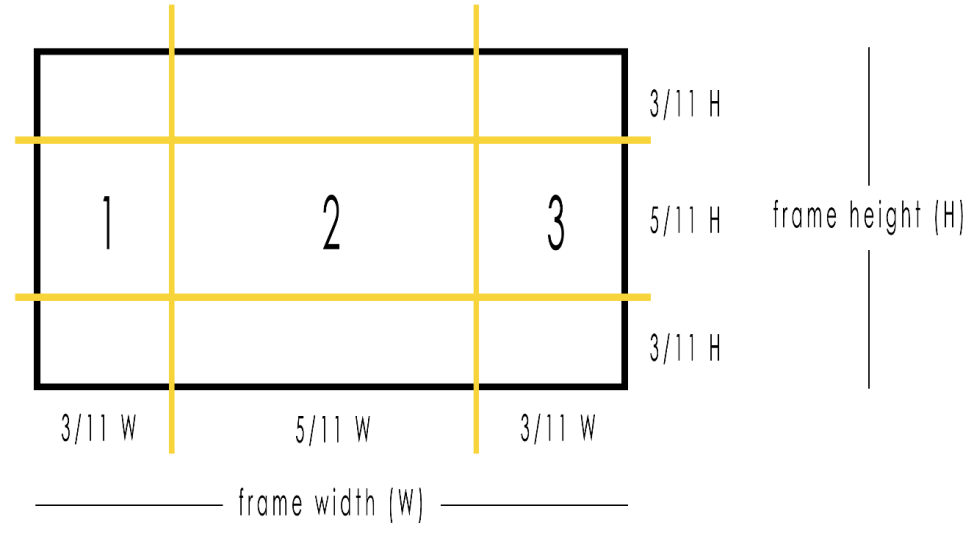
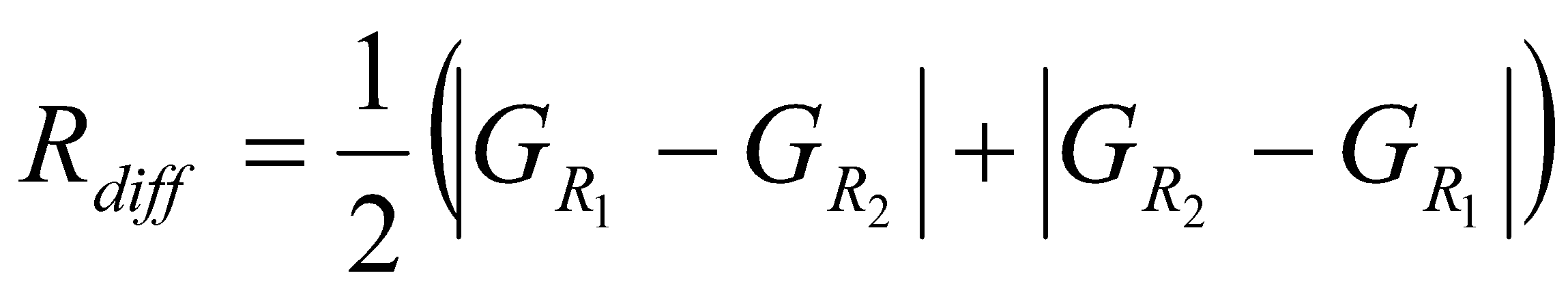


Figure 4-20 Golden Section spatial composition rule

Then applying a Bayesian classifier on the following two features:

**GR:** The grass colored pixel ratio in the second region.

**Rdiff:** The mean value of the absolute grass color pixel differences between regions I and 2, and between 2 and 3.



After implementing this approach we didn’t see much use for GR as all values were very close and no clear thresholds to set to discriminate between medium and wide shots and so little in close shots so we revised to be just the Rdiff as a feature and used a very small CNN model as our model and the G value on the whole frame is dealt with according to thresholds:

**Algorithm**

for each frame in the dataset do

1. A grass color mask is applied on the frame (as discussed in section **4.6.1**) i.e. green is white and any other color is black.
2. If the grass ratio is less than 20% then classify as close.
3. Split into 3: 5: 3 grid.
4. Extract grass ratio in each region
5. Calculate Rdiff

train the model with the values of Rdiff and the corresponding labels.

**Flow Chart**

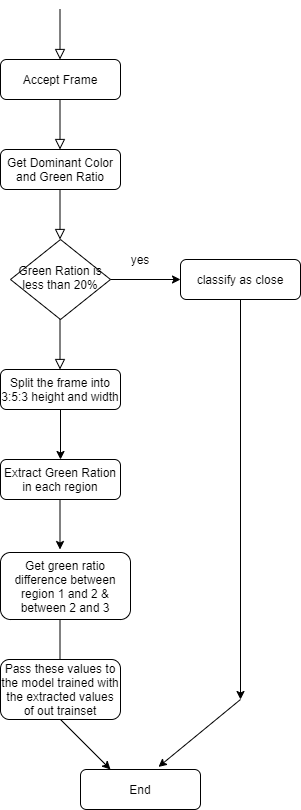


Figure 4-21 Flow Chart of shot classification

### 4.6.4. Comparison between shot classification approaches

Results of each approach are discussed in detail in chapter 6 but here we briefly compare between the above approaches in terms of classification accuracy and shortcomings

|  |  |  |
| --- | --- | --- |
| **Approach** | **Result** | **Accuracy** |
| Grass color | The grass color produces the lowest accuracy and completely fails in discriminating between an out of field shot and a logo shot as both contain very low grass ratio. | 60% |
| Face Detection | The face detection has a respectable accuracy but it's computationally very expensive as it takes nearly double the time of other approaches. | 70% |
| Image processing and machine learning | This approach had a lot of potential but the features selected were very disappointing as the model failed to discriminate between the classes properly. | 60% |
| Deep learning | As expected, the deep learning approach produced the best accuracy and also faster run time so it's the approach we use in our system. | 86% |

Table 4-1 Comparison between shot classification approaches

## 4.7. Summarization and Classification phase

### 4.7.1. Functional Description

The main purpose of this phase is to choose a number of important shots from the whole video shots, then these important shots are concatenated together to form the final summarized video.

After having the summarized video, we will classify the sequence of these important events.

The sequence of each important shots will be classified into 3 classes:

1. Goal Event
2. Attack Event
3. Other Event (fouls, offsides, substitution, etc.)

### 4.7.2. Modular Decomposition

#### 4.7.2.1. Extracting Important Events

Till this point we have some important information about each shot:

1. Shot frame number, shot start time, shot end time →output of **Shot Boundary**
2. Shot type → output of **Shot Classification**
3. Whether a goal is scored or not → output of **Goal detection**
4. Whether the shot contains a goal mouth or not → output of **Goal Mouth Detection**
5. Whether the shot contains high volume level → output of **Audio Processing**

Every replay in the match represents an important event such as, a replay for a scored goal or an attack or even a foul, so starting by detecting the replay shots (two logo shots ), we can end up with a summarized video containing the most interesting events in the match, this process will start by detecting two consecutive logo shots, then because the first wide shot before the first logo is the shot that most likely contains the important event, we traverse backward till the first wide shot before the first logo shot, finally, having theses set of important shots we append it to the output video. We start the same process again on the remaining shots.

****

Figure 4-22 example of replay detection

**Methodology**

**Input:** shots array, each has(Shot frame number, shot start time, shot end time, Shot type,

Whether a goal is scored or not, Whether the shot contains a goal mouth or not, Whether the shot contains high volume level)

**Output:** Summarized video containing only the important shots

**Algorithm:**

1. Logo count = 0
2. for each shot in the shots array
3. if shot type == logo:
   1. logo count +=1
4. if logo count == 2:
   1. append to the output video shots
   2. if shot type == wide and logo count == 0
      1. go to 3
   3. if the shot type == logo
      1. logo count -= 1
   4. move to the backward shot
   5. go to 4
5. repeat until all shots are processed
6. End up with the final summarized video

**Flow chart:**

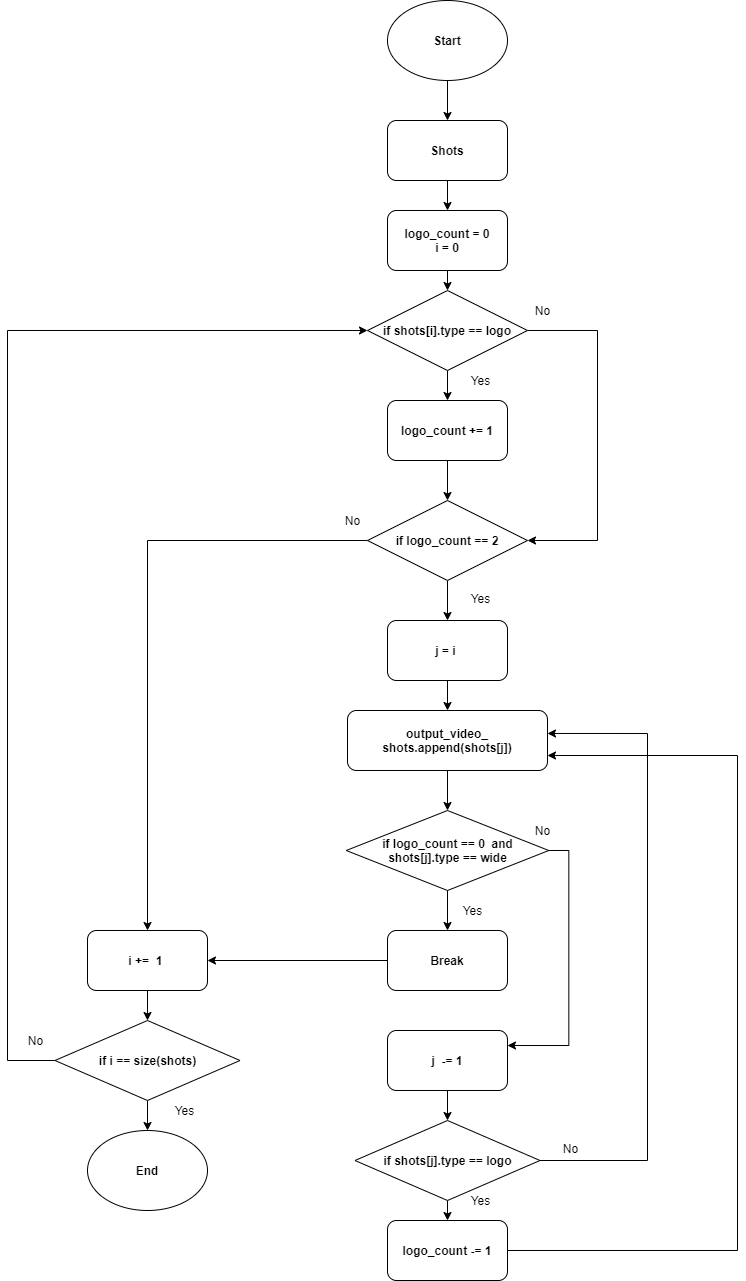
****

Figure 4-23 Flow chart of Extracting Important Events using replay detection

After detecting the important events based on the replay detection we will try to find out important events in the match that are not replayed, we will depend on the output of **Audio Processing**, for each shot we know whether each shot contains high volume level or not, which is a strong evidence of whether a certain shot is important or not.

By appending the shots that contain high audio level but not included in a replay sequence, we end up having all the important events in the match.

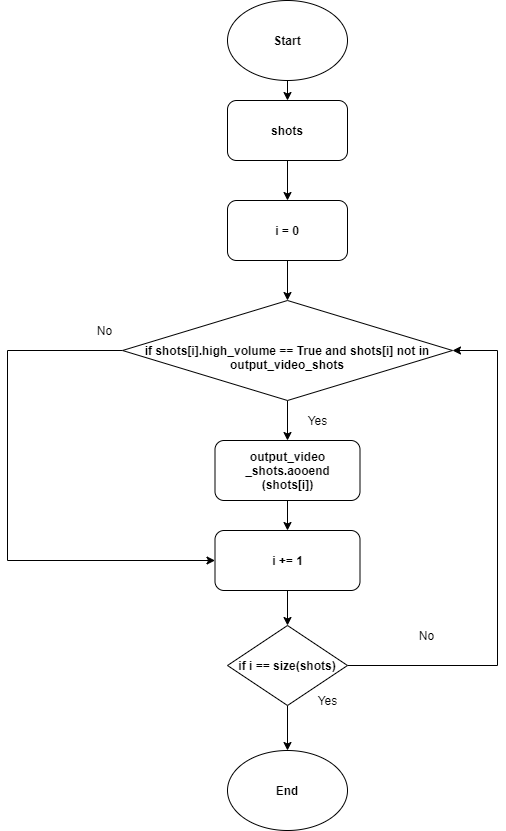


Figure 4-24 Flow chart of Extracting Important Events using audio processing

#### 4.7.2.2. Shots Sequence Classification

As we have two sets of important events:

1. Events come from replay detection
2. Events not replayed but have high volume level

We will classify each set individually, then collect the important events together and sort them based on the shot frame number, so that we have all the important shots from both sets in the right order.

**Classifying shots that come from replays**

* When a goal is scored, the score in the score board is changed which can be detected by the **Goal Detection** phase and results in “Goal Event”.
* All of the attacks are most likely to happen near the goal mouth, so finding a goal mouth in a sequence of important shots and the scoreboard is not changed, will result in an “Attack Event”.
* Other important events such as fouls, offsides, substitutions, etc. that neither contain scoreboard change, nor happen near a goal mouth will result in ”Other Event”.

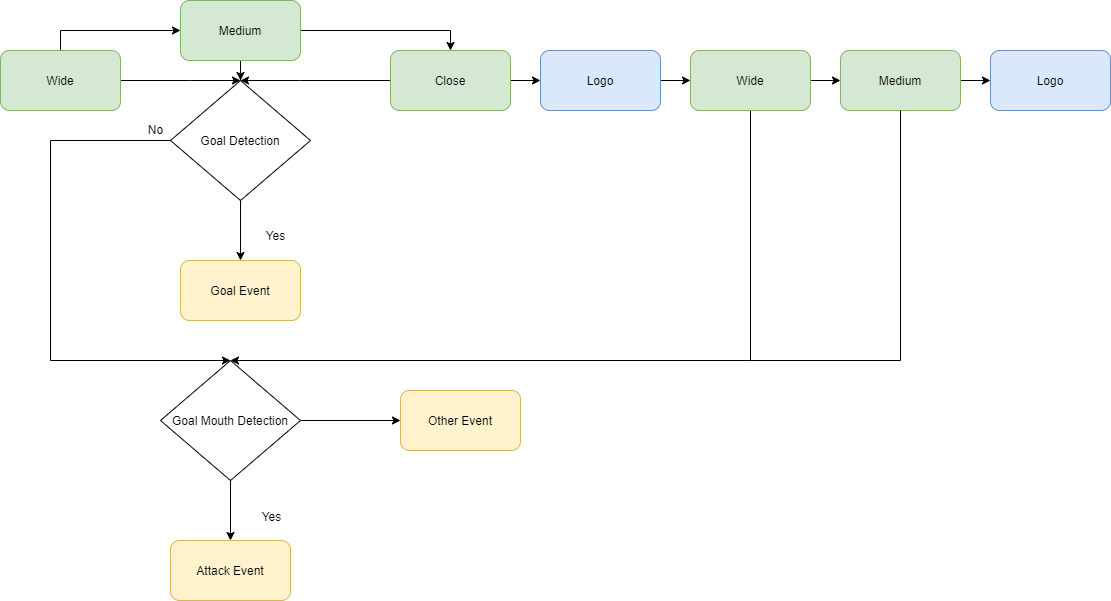


Figure 4-25 Flow of Classifying shots that come from replays

**Classifying shots that are not replayed but contain high volume level**

In classifying this set of shots we are sure that we cannot end up with a goal event as every scored goal in football matches is replayed for several times, so we have already obtained it in the replayed shots, for this set of shots we will classify the sequence of the shots in only two classes, attack event and other event, our classification will be based on detecting a goal mouth in the shot, as said before, all attacks most likely to happen near a goal mouth so detecting a goal mouth is a good indicator that a certain shot is an attack, and if a goal mouth is not found, the shot will be classified as other.

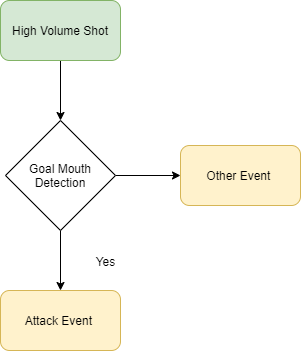


Figure 4-26 Flow of Classifying shots that are not replayed but contain high volume level

## 

## 4.8. Audio version of LAخSLY

In this section we present a version of laخsly where the summarization is only based on the audio feature i.e. and the shot boundary process i.e. when the audio level is high that gives an indication of an important event in the football match so we extract the shot in which that important event happened

### 4.8.1. Reason to develop this version

The initial results of the main version showed that theaudio feature is contributing significantly in the summarized final video so making a version with audio is worth the effort and would be much faster than the original version, with that said it’s not without its drawbacks

### 4.8.2. Advantages and disadvantage

|  |  |
| --- | --- |
| **Advantages** | **Disadvantage** |
| Depends only on the Audio and the shot boundary modules only but the main version has more modules to take into considerations. | The summarized output of this version is a relatively less accurate version as in many matches the audiences sometimes have loud voices while cheering even if it’s not really an important event in the game. |
| Much less computations compared to the main version. | some important events are not included because the audio level is not larger than the audio threshold |
| Faster runtime |  |

Table 4-2 Advantages and disadvantage of Audio version of LAخSLY

### 4.8.3. Methodology

In this version of laخsly we get the loudest moments in the match using the previously explained Audio processing module then for each loud moment in the match we index the exact frame of that video that corresponds to the loud moments then we use the shot boundary module backwards and forwards starting from that frame to get its shot. After completing that process for each loud moment, we get a collection of shots which corresponds to the loudest moments in the match. We combine those shots to create the final video

**Input:** Video Clip

**Output:** Summarized video clip

**Algorithm:**

1- Read video clip

2- Get Peaks of the audio levels using **Audio Processing module**, peaks [i…n]

3- For each peak in peaks[i], get its frame number

4- For each frame get its shot using the shot boundary module.

1- Starting from this frame explore frames descending tell finding a shot cut = shot start

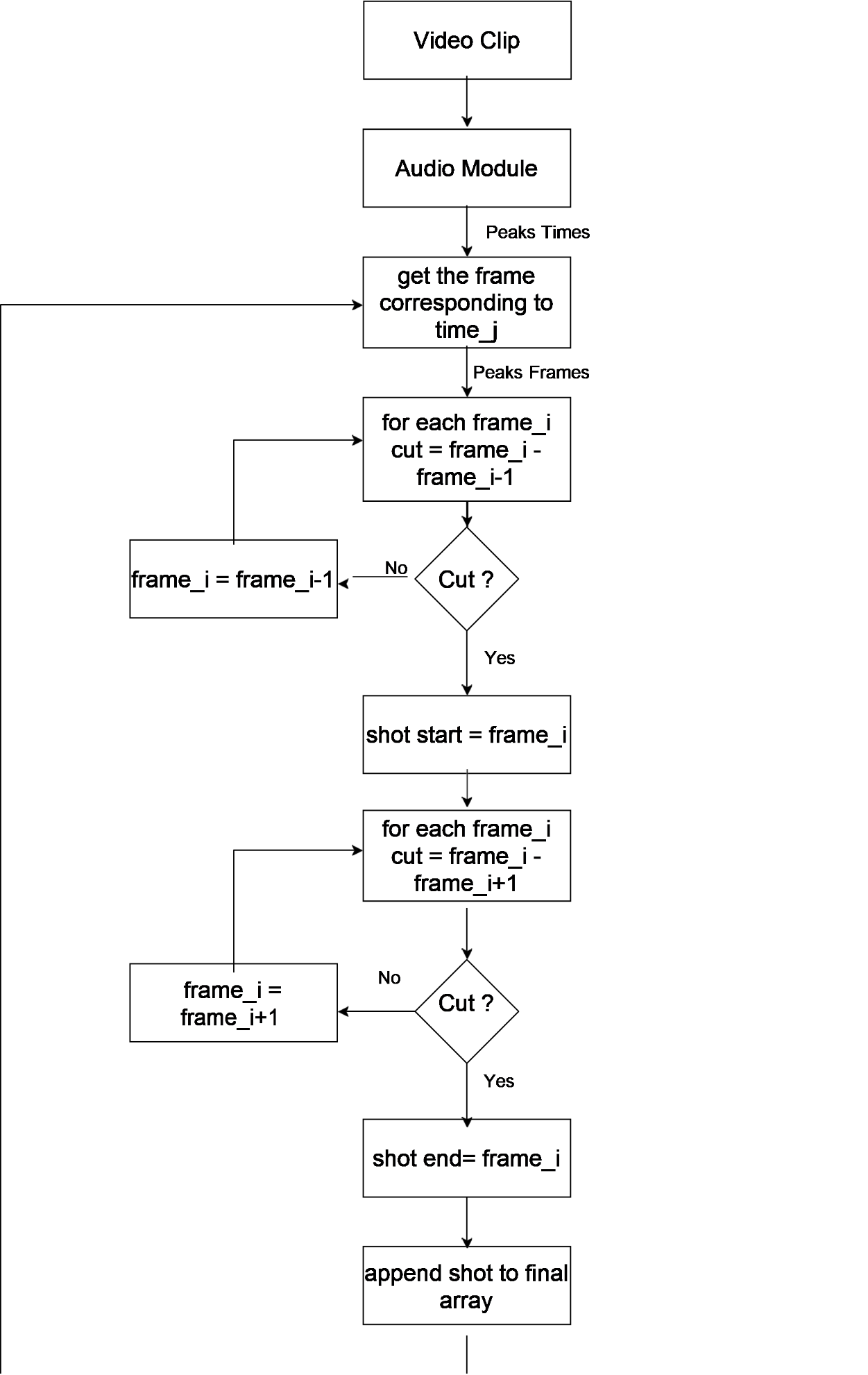
2- Starting from this frame explore frames ascendingly tell finding a shot cut = shot end

3- Having shot start and shot end, append this shot to the final array

5- Concatenate shots into one array

6- Extract the summarized video

**Flow Chart**

****

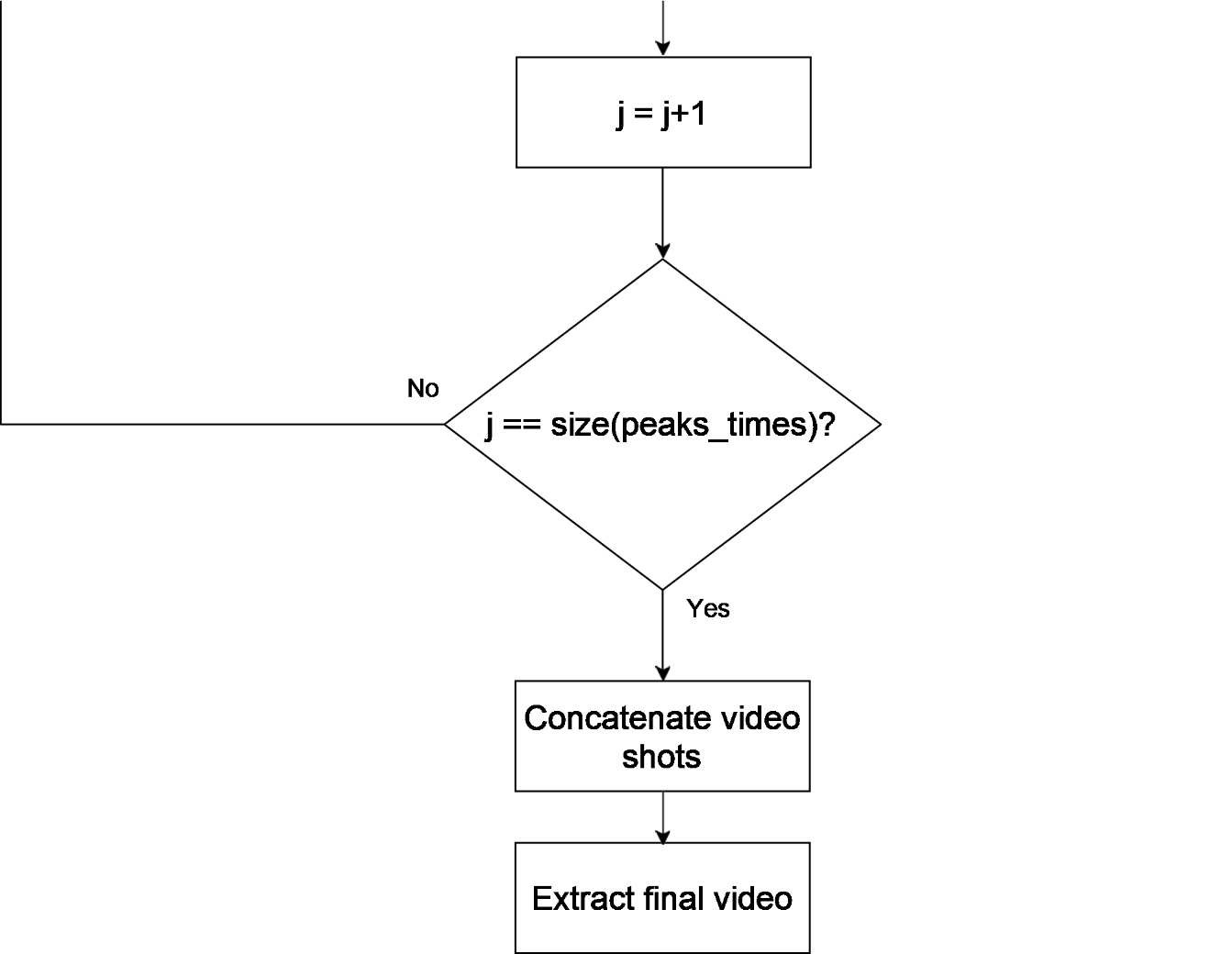
****

Figure 4-27 Flow Chart of audio processing

# Chapter 5: System Testing, Verification and Results

Testing a complete system that has a lot of separate modules each has a functionality and some modules depend on the output of others is not only a difficult task but also illogical. So, Testing each phase individually was the logical step to take, and of the one with the most importance is the shot boundary phase because all other phases depend on the results of the shot boundary phase and also to ensure that each phase is working properly on its own before integrating all phases and testing the completed system.

## 5.1. Testing Setup

To test such a system where the input is a video of a football match and the output is also a video but it’s the summary of said match, you have to have some kind of labels to verify the output whether that label is for each phase or a complete summary to test the whole system. Unfortunately, the labels are not available in our case so we had to resort to some manual verification methods and visual observation in almost all modules to ensure that they are working properly.

There is no existing way to measure the quality of a football match summary against the important events that happened in the match so we developed our own score functions to determine whether the output of the system is satisfactory or needs improvement.

### All matches tested below are from the 2018/2019 English “premiere league” and some other matches from the French league “league one” and other from the German league” Bundesliga” for testing the best audio level in the audio version of Laخsly

All results below are obtained using a personal computer with the following specifications:

* 16 Giga byte random access memory (RAM)
* Intel I7 9750 central processing unit (CPU)
* Solid state drive storage (SSD)

## 5.2. Testing Plan and Strategy

We developed a two score functions based on the goals detected, the interesting events detected and also the length of the output summarized video, goals is the most interesting event so we multiply the ratio of the detected goals and total number of goals in the match by 0.6, for other interesting events we multiply it’s ratio by 0.3 and we take the difference of 1 and ratio of length of output video and total length of video and multiply it by 0.1 to give a higher score for smaller length output video, then we add results to obtain the score.

The first score function is for detection accuracy i.e. how many important events are detected:

Detection Score = 0.5\***E1** + 0.2\***E2** + 0.2\***E3** + 0.1 \* (1 - **E4**)

**E1**: no. Goals detected / total goals

**E2**: no. of attacks detected / total no. of attacks

**E3**: no. of other events detected / total no. of other events

**E4**: Length of output video / total length of video (=90)

### The second is for classification accuracy i.e. how many events are classified correctly:

Classification Score = 0.5\***K1** + 0.25\***K2** + 0.25 \* **K3**

**K1**: no. Goals correctly classified / total goals

**K2**: no. of attacks correctly classified / total no. of attacks

**K3**: no. of other events correctly classified / total no. of other events

### After testing each module according to its respective function, the complete system would be tested as follows as follows: Each football match tested would produce a detection score and a classification score. The Average of these scores would be the final accuracy of the system. Both scores range from 0 to 1 and 1 means that the system is performing appropriately i.e. for both scores the higher the better.

### 5.3 Module Testing and Results

#### 5.3.1 Shot boundary testing and results

The shot boundary algorithm was the first module developed and as mentioned before it is the most important because all other phases except the summarization depend on its output therefore it required a lot of testing and verification.

Unfortunately, there is not a source where a person can acquire a football match and the start and end of each shot in the match, that kind of information is not available. At least for free so we had to resort to visual observation to verify the shot boundary algorithm output.

Extracting the video frames into disk and observing the cuts then comparing the output of the algorithm with what we see in the frames was the simplest and fastest way to improve the algorithm and refine the values of the thresholds used in it. The results were far better from what is anticipated

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Duration in minutes | Number of cuts | Efficiency |
| Test1.mp4 | 5 | 15 | 100% |
| Test2.mp4 | 2:22 | 21 | 80.9% |
| Test3.mp4 | 1:36 | 19 | 89.4% |
| Test4.mp4 | 3:12 | 38 | 84.2% |
| Test5.mp4 | 3:18 | 38 | 92% |
| Test6.mp4 | 4:41 | 37 | 86% |

Table 5-1 Shot boundary testing and results

#### 5.3.2 Goal-Mouth testing and results

For testing the algorithm approach a dataset of 100 wide shot frames and 100 medium shot frames which contains frames with goal-mouth and frames without goal-mouth is used for evaluation.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Wide shot** | **Medium shot** | **overall** |
| **No. Of frames** | **100** | **100** | **200** |
| **Correct output** | **87** | **81** | **168** |
| **Accuracy** | **87%** | **81%** | **84%** |

Table 5-2 Goal-Mouth testing and results

#### 5.3.3 Shot Classification using Deep Learning approach testing and results

For testing the technique, a test set of 100 images contain all classes and evaluate the results.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Image Class | Number of images | Success | Wrong | Accuracy |
| close out | 16 | 13 | 3 | 81.25% |
| close | 38 | 19 | 19 | 50% |
| wide | 22 | 22 | 0 | 100% |
| medium | 19 | 17 | 2 | 89.5% |
| logo | 20 | 20 | 0 | 100% |

Table 5-3 Shot Classification using Deep Learning approach testing and results

#### 5.3.4 Goal Detector using SSIM approach

The goal detector module was presented with 6 different matches with different scores

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Match | Score | Success | Wrong | Accuracy |
| Man City VS Brighton | 4 - 0 | 4 | 0 | 100% |
| Crystal Palace VS Liverpool | 1 - 2 | 3 | 0 | 100% |
| Liverpool VS Brighton | 2 - 1 | 3 | 0 | 100% |
| Liverpool VS Everton | 5 - 2 | 7 | 0 | 100% |
| Aston Villa VS Liverpool | 1 - 2 | 3 | 0 | 100% |
| Sheffield United VS Burnley | 3 - 0 | 3 | 0 | 100% |

Table 5-4 Goal Detector using SSIM approach testing and results

#### 5.4 Audio version testing and results

The purpose of this version is to be faster than the main version which it certainly is bus as demonstrated in following section it’s not as accurate in detecting all important events and that is indicating by the detection score function. The classification score is not applicable here as Audio only would not differentiate between a goal and an attack.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Match | Half | Output (minutes) | Goal  Detection | Attack Detection | Other Event Detection | Run Time  (minutes) | Score |
| Man City vs Brighton (4-0) | 1st | 7:41 | 2/2 | 8/17 | 2/6 | 2.6 | 0.75 |
| 2nd | 6:42 | 2/2 | 7/16 | 1/7 | 2.3 | 0.71 |
| Liverpool vs Brighton (2-1) | 1st | 4:30 | 2/2 | 10/22 | 1/9 | 1.65 | 0.71 |
| 2nd | 5:05 | 1/1 | 9/19 | 3/12 | 1.7 | 0.74 |
| Sheffield United vs Burnley (3-0) | 1st | 5:35 | 3/3 | 8/17 | 1/3 | 1.78 | 0.75 |
| 2nd | 8:54 | 0/0 | 11/19 | 3/11 | 2.8 | 0.76 |

**Avg Detection Score = 0.74**

**Avg Run time = 2.13 minutes**

While testing the audio version we performed a side experiment to see which audio level is best used in the algorithm in different leagues around the world so, to find out which audio level is best for each league we had to try different matches for different leagues around the world with the three levels to determine which is the best suited for its respective league to produce the best summary.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **League** | **Match** | **Audio Level** | **Output video Length in minutes** | **Goals Detected accuracy** | **Other events detection accuracy** | **Score** |
| BUNDESLIGA | Fc Augsburg  VS  Dortmund | 85 | 28:06 | 4/8 | 7/13 | 0.53 |
| 90 | 10:03 | 3/8 | 3/13 | 0.38 |
| 95 | 8:51 | 2/8 | 0/13 | 0.24 |
| Bayern Munich  VS  Bremen | 85 | 16:06 | 6/7 | 16/16 | 0.9 |
| 90 | 12:07 | 5/7 | 14/16 | 0.78 |
| 95 | 8:28 | 4/7 | 11/16 | 0.64 |
| Bayer Leverkusen  VS  Dusseldorf | 85 | 20:35 | 3/3 | 16/16 | 0.98 |
| 90 | 15:50 | 3/3 | 10/16 | 0.87 |
| 95 | 6:47 | 2/3 | 6/16 | 0.6 |
| LIGUE 1 | PSG  VS  Monaco | 85 | 31:14 | 6/6 | 9/12 | 0.89 |
| 90 | 23:46 | 6/6 | 10/12 | 0.92 |
| 95 | 14:48 | 5/6 | 7/12 | 0.76 |
| Rennes  VS  Nantes | 85 | 17:28 | 5/5 | 11/14 | 0.92 |
| 90 | 12:38 | 4/5 | 8/14 | 0.74 |
| 95 | 5:01 | 4/5 | 4/14 | 0.66 |
| PSG  VS  Marseille | 85 | 26:27 | 4/4 | 8/10 | 0.91 |
| 90 | 19:44 | 4/4 | 7/10 | 0.89 |
| 95 | 8:35 | 4/4 | 6/10 | 0.87 |
| PREMIER LEAGUE | Man. United  VS  Man. City | 85 | 29:38 | 2/2 | 5/8 | 0.85 |
| 90 | 21:42 | 2/2 | 4/8 | 0.83 |
| 95 | 10:33 | 2/2 | 2/8 | 0.76 |
| Aston Villa  VS  Man. City | 85 | 25:16 | 7/7 | 8/8 | 0.97 |
| 90 | 17:01 | 7/7 | 7/8 | 0.94 |
| 95 | 13:07 | 7/7 | 4/8 | 0.83 |
| Chelsea  VS  Arsenal | 85 | 24:29 | 4/4 | 7/10 | 0.88 |
| 90 | 18:04 | 4/4 | 6/10 | 0.86 |
| 95 | 6:51 | 3/4 | 3/10 | 0.63 |

Table 5-5 Audio version testing and results

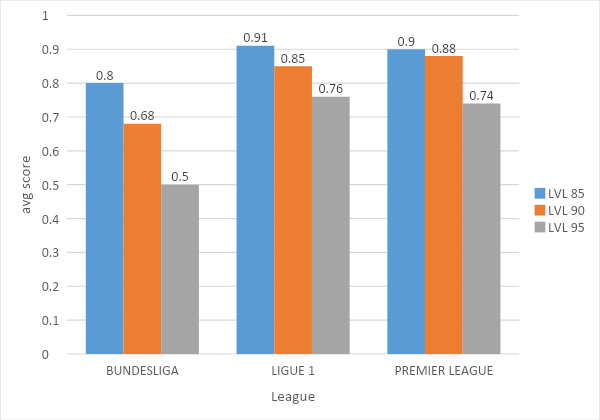


Figure 5-1 chart showing the average score for each league in each audio threshold level

In the above chart, for each league an average score is calculated for each audio threshold level by adding their scores and dividing it by total number of matches (=3).

For most leagues we have tested, level 85 threshold has the best results.

In premier league level 90 threshold has an average score very close to level 85 threshold so we can use it instead to reduce the length of the output summarized video.

### 5.4 System Testing

**5.4.1 Detection Score and run time**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Match | Half | Summary (minutes) | Goal  Detection | Attack  Detection | Other Detection | Run Time  (minutes) | Score |
| Man City vs Brighton (4-0) | 1st | 14:34 | 2/2 | 17/17 | 6/6 | 6.84 | 0.98 |
| 2nd | 11:18 | 2/2 | 15/16 | 7/7 | 5.7 | 0.97 |
| Liverpool vs Brighton (2-1) | 1st | 14:39 | 2/2 | 22/22 | 9/9 | 7.7 | 0.98 |
| 2nd | 12:41 | 1/1 | 19/19 | 12/12 | 7.5 | 0.98 |
| Sheffield United vs Burnley (3-0) | 1st | 15:08 | 3/3 | 17/17 | 3/3 | 7.11 | 0.98 |
| 2nd | 12:33 | 0/0 | 19/19 | 11/11 | 7.15 | 0.98 |

**Avg Detection Score = 0.98**

**Avg Run Time = 7 minutes**

From the results above we can clearly see the difference between the audio version and the main version as the detection score of the main version is near perfect but its run time are larger.

**5.4.2 Classification score**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Match | Half | Goal Classification | Attack Classification | Other Classification | Score |
| Man City vs Brighton (4-0) | 1st | 2/2 | 15/17 | 5/6 | 0.92 |
| 2nd | 2/2 | 9/16 | 6/7 | 0.85 |
| Liverpool vs Brighton (2-1) | 1st | 0/2 | 13/22 | 7/9 | 0.34 |
| 2nd | 1/1 | 16/19 | 10/12 | 0.91 |
| Sheffield United vs Burnley (3-0) | 1st | 3/3 | 12/17 | 3/3 | 0.93 |
| 2nd | 0/0 | 17/19 | 9/11 | 0.93 |

**Avg Classification Score: 0.81**

The classification accuracy (score) is lower than the detection score because the classification depends on the accuracies and outputs of the previous phases so any incorrect output from a previous phase will cause the classification to be incorrect. And the detection is higher because the system is not dependent on one source of information only to determine whether the shot is important or not, for example, we append shots that contain high volume but not replayed to the final output video to overcome any error might happen as a result of not detecting a replay, Although we depend on two sources to determine the importance of the shot to overcome any bad accuracy of the previous phases, but if more errors occur from several phases will badly affect this score.

## 

## 5.5 Comparative Results to Previous Work

# 

# Chapter 6: Conclusions and Future Work

We discussed our motives, area of work, problem definition and the proposed solutions. We provided the necessary background knowledge for each chapter to make the reader able to read through them and understand everything flawlessly. After that we discussed the details of our approach. We presented each phase of the system in detail starting with Pre-Processing then shot boundary detection, then the different techniques used in shot classification both image processing techniques and machine learning techniques to classify the shots obtained from shot boundary detection into five classes: Wide, Medium, Close, Close-out, and logo. Logo detection is then used to detect replay segments which indicate that an interesting event happened. We discussed the overall fundamentals for the event detection phase in which we extract features such as high levels of audio, goal-mouth detection, and goal detection. Finally, we concluded the summarization phase in which we combine all the information obtained from the all preceding phases to produce a high-quality summary of the input match.

## 6.1. Faced Challenges

* Grass color Detection: Detecting the degree of green color of the field can be difficult due to many environmental conditions such as (weather, sunset, snow and rain). Also, the stadium condition can cause some challenges as well such as (lighting and shadows). Also, the field properties itself can have a major effect as well in detecting the green region.
* Shot boundary Detection: Detecting shot boundaries is to be able to identify shot transitions which can be affected by many variables such as a fast movement of the camera or players which causes a wrong detection of a cut which in turn completely ruins that specific shot in the next phases of the system.
* Shot Classification: As shown in chapter 4, we tried a lot of approaches in shot classification searching for the best accuracy possible. Most research papers apply image processing techniques to classify frames of football match which leads to a moderate accuracy (60%) and this accuracy didn’t suit our desirable performance, so segmented matches and categorized the frames into the five classes (wide, medium, logo, close, close-out). Every class is about 8,000 images and this is good for using a deep learning model instead of image processing techniques. After training the model with the constructed data, the model suffered from over-fitting but we overcame this problem by adding drop-out regularization in some layers with probability 0.6 and fortunately the results got better.
* Goal Mouth: Research papers that use image processing to detect goal-mouth don’t deal with different frame types of shots, especially medium shots or different camera angles. Also, they don't give enough details on how they achieved their proclaimed results using those techniques. Many of the image processing techniques like canny edge detection and Hough transform have many thresholds which make it difficult to find the ideal combination of thresholds values to minimax the accuracy. Using trial and error to find the ideal set of thresholds values combination for better accuracy and trying different approaches and different image processing techniques from many research papers to find out what works better for each shot type.
* Goal detection: In goal detection we proposed two solutions, the first is using OCR and that is efficient but very slow. So, we searched for ways to compare images and be as efficient as possible until we found the second solution which uses image the structural similarity index (SSIM) of the image instead of differencing the two images or extracting the number of the score.
* Summarization and classification: It is the integration phase between the output of all previous phases, so the main challenge was that the accuracy and the output of this phase depends on the accuracies and outputs of the previous phases so any incorrect output from previous phases will cause the output of this phase to be incorrect. we tried to tackle this problem by not depending only on one source of information to determine whether the shot is important or not, for example, we append shots that contain high volume but not replayed to the final output video to overcome any error might happen as a result of not detecting a replay, Although we depend on two sources not only one to determine the importance of the shot to overcome any bad accuracy of the previous phases, but if more errors occur from several phases will badly affect this phase.

## 6.2. Gained Experience

The researching process was exhaustive but ultimately beneficial as reading many research papers and implementing many algorithms in search for the best accuracy or performance is fulfilling.

Working on a project like this guarantees valuable experience and precious knowledge in the fields of computer vision, machine learning and image processing. Researching to find a solution for a hard problem such as shot boundary, goal mouth or shot classification is experience in itself because you can apply the research process to any problem irrelevant of the field.

## 6.3. Conclusions

Laخsly is a fast and accurate football summarization system that consists of five phases, pre-processing phase in which we divide the video in a number of patches each patch is roughly 2000 frames to avoid memory overhead.

shot boundary detection phase segments the whole input video stream into small video shots by applying a combination of image processing techniques and histogram comparison techniques.

In the shot classification phase , the system applies different algorithms, namely; Grass ratio, Face Detection ,Deep learning and image processing coupled with machine learning techniques to classify a shot into one of the following classes :wide – medium - close – close out – logo and use the order of shot to detect an important event

In The event detection phase each shot is considered separately and compared with the previous shot to determine if a goal has happened between the two shots by comparing the scoreboard in the upper left corner. Two methods are proposed to detect a goal event, Optical character recognition (OCR) and structural similarity image index (SSIM). audio loudness is detected as well as the goal mouth

Lastly, the summarization phase which takes into account the output of the previous phases to decide if the shot is important and should be included in the final video or should be discarded. For example, if a shot contains a goal then it is important or if a shot has a goal mouth and the audio level is high then it is important or it’s a replay shot and a replay of course is for something important. Then combining the important shots only in a video and that is the output of the system.

There is an even faster version of laخsly that relies on shot boundary and audio loudness only to produce a summary of the match in record time.

There are a few limitations that are very hard to overcome in any sports video summarizations such as the sheer volume and variety in leagues and tournaments around the world which makes it very hard to create a global detection method for the score for example or a find a uniform way to detect important event therefore you have to tailor your system to a finite set of league that have common features.

## 6.4. Future Work

There are a few ways that are worthy of study and research to improve upon the existing system such as:

* Performance is not really an issue as the current version is very fast even on an average computer but taking advantage of a distributed systems would certainly be an improvement and would allow for producing a lot more summaries
* Make logo classification work with more leagues and tournaments by creating more datasets from those leagues.
* Expanding the services provided by the website to be more user driven where users can share and upload their own summaries.
* laخsly is created as a web service but developing a mobile application would certainly increase the user base and wouldn’t be hard to develop and maintain due to the already existing infrastructure.

# References

1. Yihong Gong, Lim Teck Sin, Chua Hock Chuan, Hongjiang Zhang and Masao Sakauchi, "Automatic parsing of TV soccer programs," Proceedings of the International Conference on Multimedia Computing and Systems, Washington, DC, USA, 1995, pp. 167-174, doi: 10.1109/MMCS.1995.484921.
2. Liu, B., Yeo, B., Yow, D., & Yeung, M.M. (1995). Analysis And Presentation Of Soccer Highlights From Digital Video.
3. O. Utsumi, K. Miura, I. Ide, S. Sakai and H. Tanaka, "An object detection method for describing soccer games from video," Proceedings. IEEE International Conference on Multimedia and Expo, Lausanne, Switzerland, 2002, pp. 45-48 vol.1, doi: 10.1109/ICME.2002.1035714.
4. Ekin, Ahmet & Tekalp, A. & Mehrotra, Rajiv. (2003). Automatic Soccer Video Analysis and Summarization. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society. 12. 796-807. 10.1109/TIP.2003.812758.
5. Ling-Yu Duan, Min Xu, Qi Tian, Chang-Sheng Xu and J. S. Jin, "A unified framework for semantic shot classification in sports video," in IEEE Transactions on Multimedia, vol. 7, no. 6, pp. 1066-1083, Dec. 2005, doi: 10.1109/TMM.2005.858395.
6. Histogram Comparison — OpenCV 2.4.13.7 documentation. (2020). Retrieved 28 May 2020, from <https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_comparison/histogram_comparison.html.>
7. Chung-Lin Huang, Huang-Chia Shih and Chung-Yuan Chao, "Semantic analysis of soccer video using dynamic Bayesian network," in IEEE Transactions on Multimedia, vol. 8, no. 4, pp. 749-760, Aug. 2006, doi: 10.1109/TMM.2006.876289.
8. Radha Penumala, M. Sivagami, Suresh Srinivasan. "Automated Goal Score Detection In Football Match Using Key Moments". Procedia Computer Science, vol 165, 2019, pp. 492-501., doi:https://doi.org/10.1016/j.procs.2020.01.017.
9. Wang, Z.; Simoncelli, E.P.; Bovik, A.C. (2003-11-01). Multiscale structural similarity for image quality assessment. Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers, 2004. 2. pp. 1398–1402 Vol.2.
10. Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

# Appendix A: Development Platforms and Tools

## A.1. Software Tools

Python is an interpreted, high-level, general-purpose programming language

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

MoviePy is a Python module for video editing, which can be used for basic operations (like cuts, concatenations, title insertions), video compositing (a.k.a. non-linear editing), video processing, or to create advanced effects. It can read and write the most common video formats

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow

# 

# Appendix B: Use Cases

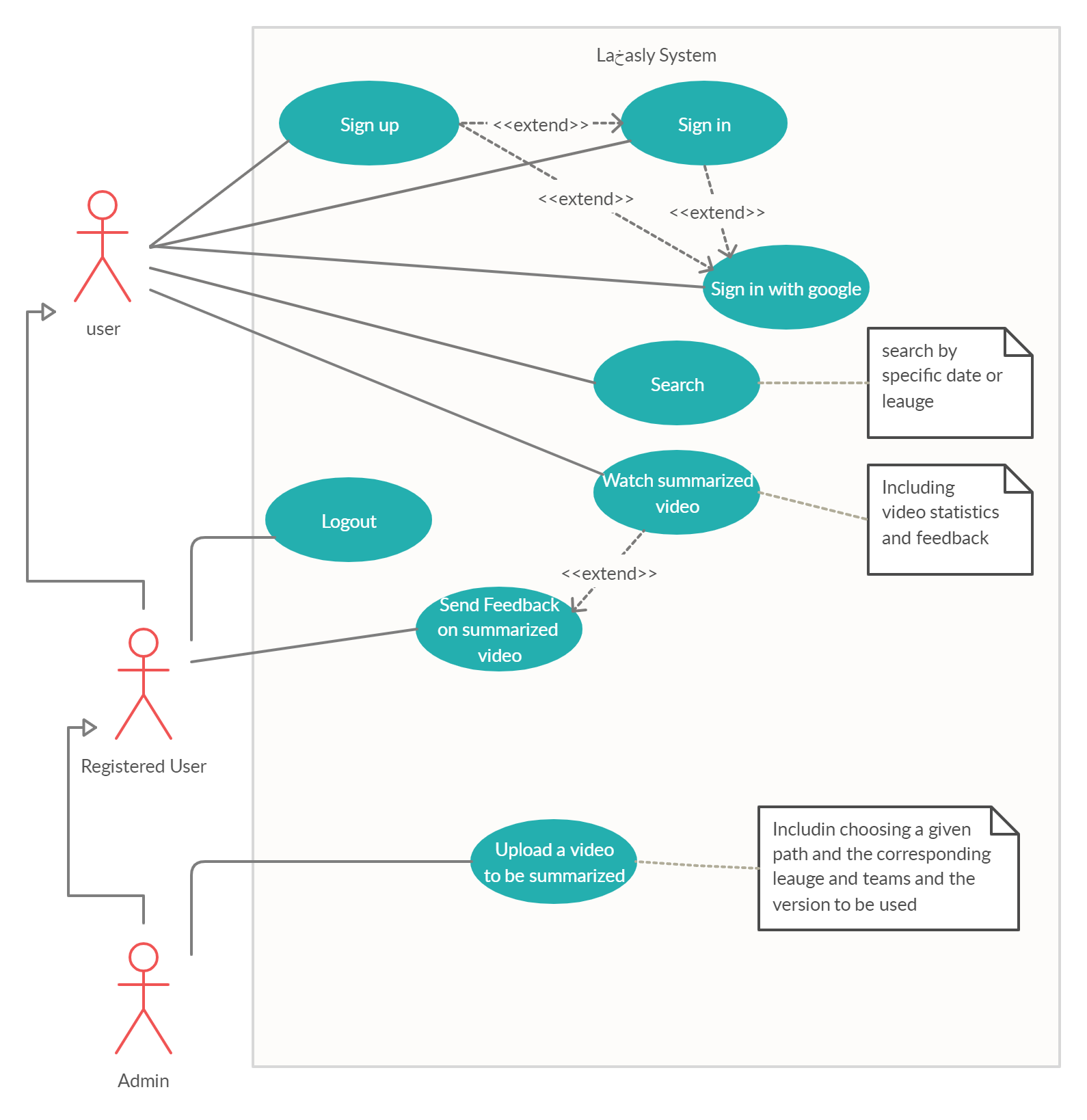


Figure A-1 Use Cases Diagram

# 

# Appendix C: User Guide